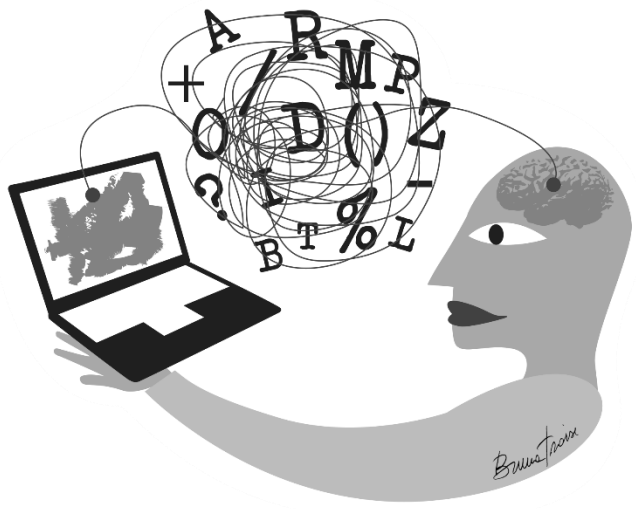
UNIVERSITÀ DEGLI STUDI DI SALERNO
Dottorato in Scienze della Comunicazione

Formal Linguistic Models and Knowledge Processing

A Structuralist Approach to Rule-Based Ontology Learning and Population

Maria Pia di Buono



Supervisor
Prof. Mario Monteleone

Coordinator
Prof. Alessandro Laudanna

XIII Ciclo – Nuova Serie
2012-2015

TABLE OF CONTENTS

Perché volar da soli è solamente un'illusione…

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor, Dr. Mario Monteleone, for his guidance, concern and advice in all academic matters. Without him none of this could have happened. Our fruitful and interesting exchange of ideas laid the basis for several considerations on research topics and also life. He may be compared to a demiurge in my research work. Thank you also for reading countless drafts of (often last minute) papers and helping me understand the Linguistic community.

I am also grateful to Prof. Annibale Elia for both his support and approval of my research ideas and proposals. He has always been very helpful and encouraging with his suggestions, comments and contributions.

I would also like to thank Prof. Alessandro Laudanna, coordinator of the Doctoral School of Communication Sciences, for his patient and support, during all the years of my doctoral research.

I am especially indebted to Max Silberztein of the University of Franche-Comté for his precious help and suggestions and for his availability in answering my several questions about NooJ, the NLP tool, on which mainly I have based my research. His input pushed me to improve my skills in using lingwares and to investigate different areas of the Computational Linguistics field and its NLP tools and resources.

I express my gratitude to Dr. Kristina (Vučković) Kocijan for the review of my thesis and for her advice, hoping to collaborate in the future.

I am also very grateful to all my colleagues and co-authors of various papers, among these, especially Prof. Johanna Monti and Dr. Federica Marano for their suggestions and friendship.

A special mention for my family and my friends, particularly Marco, which supported and encouraged me in the achievement of this important goal. They have been very understanding towards their busy Marp, especially in the last few months of thesis writing.

Finally, I should thank Sid, my punk cat, for its thoughtful contributions to this work.

SOMMARIO

L'obiettivo principale di questa ricerca è proporre un approccio strutturalista per l'elaborazione automatica della conoscenza attraverso l'apprendimento e il popolamento di ontologie, realizzata da/per testi strutturati e non strutturati. Il metodo suggerito include approcci di semantica distribuzionale e teorie di formalizzazione dei linguaggi naturali, al fine di sviluppare un quadro di riferimento che si basa su un'analisi linguistica *fine-grained*.

Partendo da una panoramica degli algoritmi di apprendimento automatico più diffusi e degli approcci basati su regole, presenteremo una metodologia per la creazione di un parallelismo tra formalismi macchina e modelli linguistici. In particolare, nella sezione 1, faremo una breve introduzione su alcuni concetti fondamentali, come la conoscenza, la rappresentazione e il ragionamento logico. Successivamente, si prenderà in considerazione la relazione esistente tra rappresentazioni formali e i linguaggi naturali e si introdurranno le norme per gli schemi di metadati e i modelli concettuali disponibili per il dominio dei Beni Culturali (BBCC).

Nella sezione 2, per affrontare i principali compiti relativi all'elaborazione automatica della conoscenza basata sulle ontologie, useremo la definizione di ontologia, richiamando anche la sua struttura e gli obiettivi.

Nella sezione 3, introdurremo alcuni dei principali metodi stocastici/statistici utilizzati per l'estrazione della conoscenza e dell'informazione attraverso ontologie. Per ciascuna delle tecniche presentate, forniremo una descrizione accurata, insieme ad alcuni esempi di applicazioni specifiche.

Nella sezione 4, in riferimento all'apprendimento e al popolamento di ontologie, introdurremo i principali modelli e i metodi utilizzati in compiti di trattamento automatico del linguaggio naturale che si basano su diversi tipi di *framework*. Infatti, al fine di analizzare le lingue naturali e storiche, l'elaborazione linguistica guida il livello di analisi, che può riguardare - contemporaneamente o separatamente - i tre diversi strati pertinenti a fonologia, sintassi e semantica.

Per quanto riguarda questi argomenti, al punto 5, proporremo il nostro approccio, basato sul quadro teorico del Lessico-Grammatica (LG), per il raggiungimento della formalizzazione del linguaggio naturale nel dominio di conoscenza dell'Archeologia. Intendiamo dimostrare come la nostra tecnica di formalizzazione linguistica può essere applicato sia al processo che al popolamento di un'ontologia di dominio, che mira a sviluppare un trattamento della conoscenza efficiente. La nostra formalizzazione linguistica si basa su un'osservazione accurata delle proprietà lessicali, e su un'appropriata registrazione dei dati linguistici di tutto il lessico e dei comportamenti combinatori delle entrate lessicali, includendo la sintassi e anche il lessico. Si differenzia dalle più conosciute tra le teorie linguistiche, come per esempio la grammatica profonda di Chomsky e le sue diverse derivazioni, che sono fortemente formali e basate sulla sintassi. Al fine di creare le principali risorse linguistiche da applicare nel nostro sistema, durante l'elaborazione linguistica, è stato sviluppato l'*Archaeological Italian Electronic Dictionary* (AIED). Inoltre, sono state create altre risorse linguistiche adatte ad applicare i vincoli semantici e ontologici che guidano le analisi linguistiche e i processi di estrazione.

Nella sezione 6, presenteremo il workflow di sistema che intendiamo sviluppare al fine di integrare le nostre risorse linguistiche in un ambiente adatto per un motore di ricerca semantico, chiamato Endpoint for Semantic Knowledge (ESK). ESK è strutturato come un endpoint SPARQL, che applica una analisi semantica *fine-grained*, basata sullo sviluppo di un modello di correlazione tra una serie di formalismi semantici per le macchine e un insieme di frasi in linguaggio naturale. ESK consente agli utenti di interrogare in linguaggio naturale una base di conoscenza, come DBpedia e Europeana, e di elaborare testi non strutturati, sia caricati dagli utenti che acquisiti on line, al fine di rappresentare e estrarre conoscenza.

Infine, chiuderemo la nostra ricerca valutando i suoi risultati e presentando possibili prospettive di lavoro future.

Keywords:
Elaborazione Conoscenza, TAL, Popolamento Ontologie, Apprendimento Ontologie, Modelli Linguistici Formali, Lessico-Grammatica.

ABSTRACT

The main aim of this research is to propose a structuralist approach for knowledge processing by means of ontology learning and population, achieved starting from unstructured and structured texts. The method suggested includes distributional semantic approaches and NL formalization theories, in order to develop a framework, which relies upon deep linguistic analysis.

Starting from an overview of the most spread machine learning algorithms and rule-based approaches, we will present a methodology for creating a parallelism between machine formalisms and linguistic models.

More specifically, in section 1, we will make a brief introduction to some core concepts, such as knowledge, representation and logic reasoning. Subsequently, we will consider the relationship between formal representations and natural languages and we will introduce standards for metadata schemata and conceptual models available for the Cultural Heritage (CH) domain.

In section 2, to deal with the main tasks related to ontological Knowledge Processing (KP), we will use the definition of ontology, also recalling its structure and goals.

In section 3, we introduce some of the main stochastic/statistical methods used to extract knowledge and information through ontologies. For each of the technique presented, we will provide an accurate description, together with some samples of specific applications.

In section 4, as for ontology learning and population, we will introduce the main models and methods used in NLP tasks and which are based on different types of frameworks. Actually, in order to analyse natural and historical tongues, linguistic processing addresses the level of the analyses, which may concern – contemporarily or separately – the three different layers of phonology, syntax and semantics.

As for these topics, in section 5, we will propose our approach, based on Lexicon-Grammar (LG) framework, to the achievement of natural language formalizations in the Archaeological knowledge domain. We intend to

demonstrate how our language formalization technique can be applied to both process and populate a domain ontology, aiming at developing an efficient and effective knowledge processing. Our linguistic formalization is based on an accurate observation of lexical properties, and on an appropriate linguistic data recording of all lexicon and lexical entry combinatory behaviours, encompassing syntax and, also, lexicon. It differs from the best known among current linguistic theories, as for instance Chomsky's deep grammar and its various offspring, which are strictly formalist and syntax-based.

The Archaeological Italian Electronic Dictionary (AIED) has been developed in order to create the main Linguistic Resources which are applied in our system during linguistic processing.

Furthermore, we create other resources suitable to the application of semantic and ontological constraints which drive linguistic analyses and extraction processes.

In section 6, we will present the system workflow we intend to develop in order to integrate our LRs in an environment suitable for a semantic search engine, called Endpoint for Semantic Knowledge (ESK). ESK is structured as a SPARQL endpoint, which will be applying a deep semantic analysis, based on the development of a matching model between a set of machine semantic formalisms and a set of NL sentences.

ESK allows users to run an NL query against KBs, such as DBpedia and Europeana, and to process unstructured texts, both uploaded by users and retrieved on line, in order to represent and extract knowledge.

Finally, we will close our research evaluating its results and presenting possible future work perspectives.

Keywords:

Knowledge Processing, Natural Language Processing, Ontology Population, Ontology Learning, Linguistic Formal Models, Lexicon-Grammar.

LIST OF FIGURES AND TABLES

Figures

Tables

INDEX OF MAIN ABBREVIATIONS AND ACRONYMS

Anaphora Resolution	AR
Archaeological Italian Electronic Dictionary	AIED
Artificial Intelligence	AI
Atomic Linguistic Units	ALUs
Atomic Linguistic Units	ALUs
Categorial Grammar	CG
CIDOC Conceptual Reference Model	CIDOC CRM
Classification and Regression Trees	CART
Closed World Assumption	CWA
Combinatory Categorial Grammar	CCG
Computer Science	CS
Coreference Resolution	CO
Cross-Lingual Information Retrieval	CLIR
Cultural Heritage	CH
Data, Information, Knowledge, Wisdom	DIKW
Decision Tree	DT
Defense Advanced Research Projects Agency	DARPA
Distributional Hypothesis	DH
Endpoint for Semantic Knowledge	ESK
Europeana Data Model	EDM
Finite-State Automata	FSA
Finite-State Transducers	FSTs
Fondazione Bruno Kessler	FBK
Formal Languages	FLs
fuzzy c-means	FCM
Generalized Phrase Structure Grammar	GPSG
Head-driven Phrase Structure Grammar	HPSG
Hidden Markov Model	HMM
Hyperspace Analogue to Language	HAL
Information Extraction	IE
Information Retrieval	IR
Information Science	IS
Knowledge Acquisition	KA
Knowledge Bases	KBs
Knowledge Extraction	KE
Knowledge Processing	KP
Knowledge Representation	KR
Language Models	LMs
Latent Semantic Analysis	LSA
Lexical Functional Grammar	LFG
Lexicon-Grammar	LG
Light Verb	LV

Lightweight Information Description Objects	LIDO
Linguistic Linked Open Data	LLOD
Linguistic Resources	LRs
Linked Open Data	LOD
Machine Learning	ML
Machine Translation	MT
Meaning Text Theory (MTT)	MTT
Message Understanding Conferences	MUC
MultiWord Expressions	MWEs
MultiWord Units	MWUs
Naïve Bayes	NB
Named Entity Recognition	NER
Named Entity Recognition and Classification	NERC
Natural Language	NL
Natural Language Processing	NLP
Neural Networks	NNs
Noun Phrase	NP
Ontology Learning	OL
Resource Description Framework	RDF
Simple Knowledge Organization System	SKOS
SPARQL Protocol and RDF Query Language	SPARQL
Suggested Upper Merged Ontology	SUMO
Support Vector Machine	SVM
Term Extraction	TE
Uniform Resource Identifier	URI
Verb Phrase	VP
Web Ontology Language	OWL

FOREWORD: A STATEMENT ON ALL PRELIMINARY AND NECESSARY PREREQUISITES TO THIS RESEARCH

Meanings which we do not retrieve generate an uncommunicative real-world knowledge.

The statement 'Meanings which we do not retrieve generate an uncommunicative real-world knowledge' perfectly introduces the core of this research project. In fact, our main motivation arises from the purpose of improving wherewithal of processing real-world knowledge by means of machines. In other words, being real-world knowledge mainly encoded into digital formats, which means being manageable by machines, is crucial developing adequate techniques of processing, suitable to retrieve meanings which belong to knowledge.

If it does not, it will fail to produce high quality outputs from the commitment of knowledge processing by means of machines, generating an uncommunicative knowledge.

As we will see in the following, the main topics of this research project are devoted to analyse the relation existing between Formal Linguistic Models and Knowledge.

Generally, in Computer Science, terms used to indicate a knowledge-treatment process refer to two separate activities: knowledge representation and knowledge extraction.

As an alternative, we propose the term Knowledge Processing (KP) in order to indicate both the representation and the extraction processes, due to the fact that these activities are strictly linked. In other words, to work on only one of these is not a promising way of addressing the problem of knowledge treatment by means of machines. In fact, in our opinion, the improvement of interactions and communications between humans and machines may be achieved dealing with knowledge as far as both its representation and extraction are concerned.

Therefore, the deep core of this dissertation relates to the issue of KP, which is focused on analysing the relationship between natural languages and machine formalisms.

We propose a structuralist approach to KP, based on an accurate lexicon formalization and committed to ontology learning and population tasks. Our method, which founds on Lexicon-Grammar (LG) framework, aims at improving KP in order to demonstrate how a precise language-formalization technique can be applied to both process and populate a domain ontology.

Ultimately, we suggest a methodology for matching human knowledge, expressed into natural languages, with machine formalisms, in order to develop an environment suitable to treat text-based contents and semantic information.

Achieving these purposes entails answering some preliminary questions about what knowledge is, how and by what it is constituted, collected and stored and where we could find the *trait-d'union* between human knowledge and machine formalisms.

Generally speaking, knowledge may be intended as a *fluid mix* of tacit and explicit knowledge, which means that it is formed of different, but interacting, constituent elements. These elements, namely *data, information* and *human experience*, are continuously updated and integrated with additional knowledge, which means that they are involved in an (endless) iterative moving process. Exactly due to its not-static nature, knowledge seems to be difficult to define, and therefore it may be not analysed as a monolith, or a unique piece. It is necessary to cope with its smallest constitutive elements, which are data, information and human experience, and are all expressed by means of natural languages. In other words, being produced by humans and intended for humans, knowledge is represented, collected, extracted and communicated using natural languages. These considerations lead us to state that knowledge, intended as *Weltanschauung* (real-world knowledge), is not natively machine-readable. Consequently, for this reason, if we want to achieve a machine KP, we have to handle knowledge elements constituted by means of natural languages. During the processing of these elements, a core role is performed by the way in which we formalize human knowledge into machine formalisms. In fact, if knowledge processing requires processing of its constituent elements, expressed in natural languages, and furthermore if human knowledge is not machine-readable, then we have to apply a

representation model suitable to 'convert' natural-language elements into a machine-readable format.

On the other hand, as far as Information Sciences are concerned, natural-language elements cannot be processed as simple chains of bits, which become sequences of alphabet letters forming one or more words of a given language. Besides, it is worth saying that such words, when combining together according to specific usage rules, essentially contribute to structure complex systems in which random lexical and morph-syntactic behaviours are not expected, or even guessed. This is because all natural-language elements are featured by sets of intricate but observable characteristics, which in any language are complex to handle and strongly affect the semantic expressiveness pertaining to word combinatory meanings. In brief, in the need of using automated tools, such morph-syntactic characteristics represent something which may be inferred only by means of a pre-stablished and pre-structured deep linguistic analysis. Moreover, the real world and the machine one use two different kinds of representations, and all the considerations which follow spring from this idea, that is: to explain themselves, the real world and the machine one need different denotative elements.

Therefore, our starting hypothesis is that a coherent and consistent linguistic formal description is crucial and indispensable to achieve a correct semantic representation of a specific knowledge domain (di Buono, 2015).

Such a semantic representation has to acknowledge representational expressions, putting them in a one-to-one correspondence with the concepts in a given domain.

On such premises, ontologies seems to be the most promising means for representing semantically both Human-World and Machine-World. In fact, due to the fact that they focus on terms meaning and on the nature and structure of a given domain, ontologies are suitable to match human and machine representations. In order to justify our standpoint, firstly we introduce machine formalisms and draw upon the theory of formal languages which may be used to explain formal representations, as concerns both metadata and conceptual models.

Consequently, starting from an overview of the most spread machine learning algorithms and rule-based approaches, we will present a methodology for creating a parallelism between machine formalisms and linguistic models.

For our experiment, we will suggest an approach which aims at improving Knowledge Representation (KR) and Knowledge Extraction (KE) in the Archaeological domain, characterized by a range of variable types and properties of contents, due to the fact that Archaeological domain holds elements with a strong Semantic Expansion (SE), being strictly interlinked with other domains.

The Archaeological Italian Electronic Dictionary (AIED) has been developed in order to create the main Linguistic Resource which will be applied in our system during linguistic processing.

Furthermore, we will create other resources suitable to apply semantic and ontological constraints during the matching and extraction processes.

Finally, we will propose a system usable to integrate our LRs in an environment apt to work as a semantic search engine, which will be called Endpoint for Semantic Knowledge (ESK). ESK will be structured as a SPARQL endpoint aiming at applying a deep semantic analysis, based on the development of a matching process among machine semantic formalisms and NL sentences. ESK will allow users to run an NL query against KBs such as DBpedia and Europeana, and process unstructured texts, both uploaded by users and retrieved on line, in order to represent and extract knowledge.

More specifically, in section 1 we will make a brief introduction on some core concepts, such as knowledge, representation and logic reasoning. Subsequently, we will consider the relationship existing between formal representation and natural languages and we will introduce standards for metadata schemata and conceptual models available for the Cultural Heritage (CH) domain.

In section 2, to deal with the main tasks related to ontological Knowledge Processing (KP), we will use the definition of ontology, also recalling its structure and goals.

In section 3, we introduce some of the main stochastic/statistical methods used to extract knowledge and information through ontologies. For each of the technique presented, we will provide an accurate description, together with some samples of specific applications.

In section 4, as for ontology learning and population, we will introduce the main models and methods used in NLP tasks and which are based on different types of frameworks. Actually, in order to analyse natural and historical tongues, linguistic processing addresses the level of the analyses,

which may concern – contemporarily or separately – the three different layers of phonology, syntax and semantics.

As for these topics, in section 5, we will propose our approach, based on Lexicon-Grammar (LG) framework, to the achievement of natural language formalizations in the Archaeological knowledge domain. We intend to demonstrate how our language formalization technique can be applied to both process and populate a domain ontology, aiming at developing an efficient and effective knowledge processing. Our linguistic formalization is based on an accurate observation of lexical properties, and on an appropriate linguistic data recording of all lexicon and lexical entry combinatory behaviours, encompassing syntax and, also, lexicon. It differs from the best known among current linguistic theories, as for instance Chomsky's deep grammar and its various offspring, which are strictly formalist and syntax-based.

The Archaeological Italian Electronic Dictionary (AIED) has been developed in order to create the main Linguistic Resources which are applied in our system during linguistic processing.

Furthermore, we create other resources suitable to apply semantic and ontological constraints during matching and extraction process.

In section 6, we will present the system workflow we intend develop in order to integrate our LRs in an environment suitable for a semantic search engine, called Endpoint for Semantic Knowledge (ESK). ESK will be structured as a SPARQL endpoint, which will be applying a deep semantic analysis, based on the development of a matching process between a set of machine semantic formalisms and a set of NL sentences.

ESK will allow users to run an NL query against KBs, such as DBpedia and Europeana, and to process unstructured texts, both uploaded by users and retrieved on line, in order to represent and extract knowledge.

Finally, we will close our research evaluating its results and presenting possible future work perspectives.

In the following sections we will expand on the topics only hinted to so far.

I – FOUNDATIONS

Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it. When we enquire into any subject, the first thing we have to do is to know what books have treated of it. This leads us to look at catalogues, and at the backs of books in libraries.

Samuel Johnson (Boswell's Life of Johnson)

Before analysing how to deal with formal linguistic models in order to accomplish Knowledge Processing (KP), we will make a brief introduction on its core concepts, such as knowledge, representation and logic reasoning. Subsequently, we will consider the relationship existing between formal representation and natural languages and we will introduce standards for metadata schemata and conceptual models available for the Cultural Heritage (CH) domain.

1. Knowledge, Representation and Reasoning

What Knowledge is supposed to be represents a topic discussed by philosophers since the ancient Greeks. The concept of Knowledge, taken to mean *Weltanschauung (real-world knowledge),* draws not only on philosophical but also on scientific traditions. Indeed, the attempt to systematize knowledge started with Aristotle, runs along all subsequent epochs of human intellectual investigation, and is present in many philosophical and scientific masterpieces, as for instance Giordano Bruno's Art of Memory and Linnaeus' classification schema.

However, for the points which we wish to deal with in the pages that follow, it seems to us that Davenport & Prusak's definition is the most appropriate:

> *Knowledge is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information. It originates and is applied in the minds of knowers. In organizations, it often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices, and norms (1998:5).*

To us, this preliminary definition is very important, as it includes the context in which we want to place and develop our argumentation. Actually, we do consider knowledge as an integration of human acting and thinking, namely a 'fluid mix' of different elements, which involves formal knowledge and tacit knowledge. Formal and tacit knowledge are recognizable "not only in documents or repositories", therefore as formal knowledge, "but also in organizational routines, processes, practices, and norms", therefore as tacit knowledge. These two kinds of knowledge represent the foundation on which humans establish their new capabilities and information about the world and the things contained in. In order to (re)use and apply such knowledge, it is worth that it is represented and extracted in an adequate way.

Thus, we may assert that, under these views, KR becomes an integration of (is a matter of) Philosophy, Logics, Linguistics and Computational Linguistics.

In Computer Science (CS) and Artificial Intelligence (AI), such a knowledge, even if distinguished in structured, semi-structured and unstructured, is compared to facts. It means that knowledge is considered as a set of structured, semi-structured and unstructured, information concerning the real-word knowledge.

Since this assumption of equation with facts, "Knowledge Representation was largely seen as the task of managing collections of facts about the world" (Brewster & O'Hara, 2007).

Such collections of facts are the result of the integration of different knowledge elements, namely the constituents of the fluid mix.

Among these constituents, the main ones are represented by primitives that might be combined to produce more complex (sets of) elements. As we will see, in KR, these primitives could be rules, frames, semantic networks and concept maps, ontologies, and logic expressions (Vassev & Hinchey, 2011). In this sense, a primitive is a constituent element of the world we want to represent, namely a concept, which may be brought together with other

primitives. Thus, the main goal of Knowledge Representation (KR) becomes representing these primitives adequately, which means that they become suitable to manage the world and its facts.

In 1993, Davis *et al.* state that the first role of KR is to be a surrogate, "a substitute for the thing itself, that is used to enable an entity to determine consequences by thinking rather than acting, that is, by reasoning about the world rather than taking action in it" (1993:17). This statement highlights two important aspects as far as knowledge representation process is concerned: identity and fidelity. The first one is related to finding out what something is for, that is, establishing the identity between the surrogate and its telic referent in the world. On the other hand, fidelity is connected to setting the closeness of the surrogate to the real thing, in order to recognize which attributes have been represented and made explicit.

A surrogate is inevitably imperfect, since its representation of real-world entities contains simplified assumptions. It is concerned with natural objects, as well as formal objects, but a representation is always, and necessarily, different from the thing itself. As a consequence to this, representation fidelity is not ever accurate, that means imperfect surrogates are unavoidable. Anyway, the main scope of such a surrogate is describing the world and its facts, even if the description is imperfect. It means that creating a surrogate, namely applying a KR, enables us to lay the foundation of the reasoning about the world and its facts.

Indeed, reasoning unavoidably requires a representation of constituent concepts, inasmuch as

> (it) *is the formal manipulation of the symbols representing a collection of believed propositions to produce representations of new ones. It is here that we use the fact that symbols are more accessible than the propositions they represent. They must be concrete enough that we can manipulate them (…) in such a way as to construct representations of new propositions* (Brachman & Levesque, 2004).

In other words, reasoning allows to manipulate the symbols which lead to represent a collection of propositions about the world and its facts. Being more manageable and concrete than represented propositions, such symbols

and their manipulation are suitable to form the representation of new propositions, starting from the believed/initial ones.

Such a manner of proceeding characterizes the process of reasoning which may progress internally, but often it concerns things existing externally. It means that the process of reasoning, which founds our knowledge, is associated to both internal aspects/features and external ones. Therefore, we can distinguish two kinds of knowledge: an internal one, that is about the system, and an external one, about the system environment.

Nevertheless, due to its characteristics, available knowledge is often certain to be incomplete and for such reasons we are lead to infer starting from what we know. Indeed, sometimes, what we know involves a kind of knowledge not explicitly mentioned in the existing propositions. This condition causes the inference process, which leads to extracting knowledge from a set of given assumptions by means of general rules which specify representative properties of a fact. In other words, such a process is not guided by explicit knowledge, but is conducted by a reasoning process which is founded on a set of general rules, extracted from what is available. This set of general rules may be defined as a knowledge base (KB) from which we can deduce plausible, but not infallible, conclusions. Given that these conclusions are uncertain, when we realize that new generated information is imperfect, we have to admit that our initial rules may not be correct at all. In these cases, it is necessary to reconsider some of the initial rules in order to reach a different conclusion, which seems more plausible than the first one. Thus, such a reasoning process advances by means of rules definition and their application, which means drawing upon the so-called non-monotonic reasoning. Indeed, the non-monotonic reasoning is characterized by the possibility of modifying the set of initial assumptions when conclusions are invalidated by additional knowledge. In this way, adding knowledge may modify both the initial propositions and the propositions which may be or have been derived.

For these reasons, we may differentiate monotonic from non-monotonic reasoning. This differentiation pertains to Classic Logics, which states that:

> If a formula M is derived from a set of rules P, then it is also derivable from any superset of P.

However, by means of this assumption, which implies that any superset of P may derive the formula M, it is possible to state that Classical Logics proves to be inadequate. Let us suppose to have the following statements inside a KB:

- Typically birds fly
- Chickens do not fly
- Marta is a bird

The plausible conclusion is that 'Marta flies'; but if we add the information 'Marta is a chicken' to the KB, we have to conclude that 'Marta does not fly'.

In Classical Logics, we cannot represent the main rule ('Typically birds fly') adding an exception as 'Chickens do not fly', because we do not know all exceptions in advance. Furthermore, exceptions are considered negative information and in Classical Logics only positive information can be represented explicitly. Such an assumption, which means that only provable facts are true, is called Closed World Assumption (CWA) or monotonic reasoning. Therefore, in order to deduce that 'Marta flies', we have to prove that Marta is not an exception. Unless we can prove that Marta actually is not a chicken, to conclude that Marta is not an exception to the given property ('Typically birds fly'), we need non-monotonic reasoning mechanisms. The reason is that the non-monotonic reasoning allows to formalize inference rules without specifying all the exceptions to the initial rule. In this sense, we may assert that, due to its characteristics, the non-monotonic reasoning aims at representing a dynamic knowledge, that is what we have defined at the beginning as a fluid mix.

Representing such a dynamic knowledge seems to be a problem to overcome; indeed, several authors tried to address the issue. Starting from the late '70s, different theories have been proposed: non-monotonic logic (McDermott & Doyle, 1980); default Logic (Reiter, 1980); circumscription (McCarthy, 1980); autoepistemic logic (Moore, 1987)[1]. The main goal of such theories is to provide new formalisms useful to represent knowledge more

[1]In this paragraph we just cite different approaches to non-monotonic reasoning for introducing the inference process related to Open World assumptions, opposite to boundaries of CWA.

accurately, also considering the inadequateness of Classical Logics in defeasible reasoning formalization.
Generally speaking, a non-monotonic logic system may include:

- Default reasoning, by which we assume a truth unless we prove its contrary by more specific information;
- Negation-by-failure, in which we conclude that a proposition is false on the basis that there is a failure in proving it;
- Implicit CWA, when we conclude that an information is false because we do not have enough information about the entity.

Being a formalism based on inference rules, reasoning can be defined as a form of calculation over symbols that stand for propositions, not for numbers.

Representation and reasoning are relevant in order to structure complex systems useful to describe and predict behaviours. They define the basis of AI systems and, being both expressed in symbols, may be manipulated automatically for developing procedures in various tasks.

In this field, the challenge has been making an efficient and effective matching between human and machine semantics. It entails translating a natural language expression into a machine formalism without losing the meaning of such an expression.

Indeed, proceeding with the philosophical tradition of Bacon and Locke, modern scientists assert that knowledge can be considered as an edifice, and that concepts are blocks of this edifice. In this way, when we work on representation and reasoning, we are processing concepts. Concepts are expressed through words, therefore our procedures have to concern human language, which transmits such concepts (Brewster & O'Hara, 2007).

For this reason, one of the most debated topic, perceived in the attempt of matching human and machine semantics, is about the lexical precision and certainness of representations. In other words, the most spread KR techniques aim at creating adequate tools to represent meaning of and to infer from facts, dealing with human and machine formalisms. It is worth to notice that the human formalisms we refer to are natural languages, while formal models of knowledge representation stand for the machine formalisms. Due to this need of finding a formalism suitable for creating a correspondence between human

and machine knowledge representations, ontologies have been introduced and subsequently largely used.

Indeed, starting from Gruber's definition in 1992, a wide range of CS specialists, assumes ontologies as the connection medium between human world and machine world. Actually, ontologies allow drawing upon human language to model a domain of knowledge or discourse, outlining "an explicit and formal specification of a conceptualization" (Gruber, 1993). It means that ontologies may be considered representative of specific knowledge, namely concepts, due to the fact that as for KR they guarantee more fidelity than other formalisms. Such a representative fidelity is directly based on the use of machine formalisms which are closer to human ones. Thus, ontologies may be suitable for ensuring formal interoperability between human and machine semantics.

2. **A Semantic Digression**

According to Aristotle, on whose work the groundwork of this paragraph lays, the meaning of any concept is always defined for scientific purposes and in a rigorous way. Such rigorous definitions are arbitrary, as the results of an agreement between parties, i.e. those to whom a given concept has a crucial importance. In our world delineation, we use words in order to identify and define the world itself and its constituents, but the more we become rigorous in our definitions, the more we separate from what we may define natural. In other words, during the definition process of a concept, we are leaded up to respond to more specific questions in order to take in the field in which the concept is applied. The agreement about how to reply to these responses becomes a complex system, built arbitrarily. Such an arbitrary system is developed on a process which leads to define categories[2] and their individuals by means of a definition process comparable to the development of an ontology. Indeed, the way in which we progress during the definition process allows to classify concepts (classes) and to identify their representative words (individuals). In this way, the classification of individuals into species and

[2]Starting from this point, in our dissertation we prefer to use the term 'class' to identify a group of individuals, instead of 'category', so that the proximity with ontology definition is more clearly deduced.

genera has been started, laying the foundations of modern descriptive sciences, as taxonomic botany and zoology.

Aristotle also differentiates essential properties of a class from their accidental ones. The first ones stand for the main properties an element must have to become an individual belonging to a certain class, while the second ones are not relevant for the identification process.

Obviously, a formalized definition of classes and individuals brings about some issues, which are to be coped with and analysed. For example, we identify a particular individual, a dog called Mizar, as an instance of one corresponding class or universal kind, i.e. dogs or mammals, due to the fact that Mizar has all properties which correspond to the necessary properties of such class or kind. If one of these properties, e.g., the fact of having four legs and/or a tail, fails or lacks, could still Mizar be considered an individual of the dog kind? This (unanswered) question states the problem of the correspondence between an individual and a category, and, reasoning on a less abstract level, between a common noun, or a simple word, and a concept. It is hard to solve this issue without specifying further properties. For this reason, during the definition process, we are leaded to state that features and qualities are not to be considered as equal elements. Indeed, features refer to extrinsic properties of an individual and they can be measured or observed, e.g., the colour or the flavour of something. Aristotle defines features as accidental properties of a class; indeed, they may change without altering the essence of individuals (a dog with three legs is still an individual of the specific universal kind). On the contrary, a quality represents an intrinsic ad inherent characteristic, which outlines a certain kind of individuals, distinguishing these from others (i.e. the density property distinguishes a matter from others and it does not change regardless of how much you have of the substance). Therefore, qualities are essential properties that we refer to as necessary, but not sufficient, conditions for considering an individual as an element of the class we are defining. Thus, defining a class and its individuals entails a process of identification and definition of both features and qualities. In other words, in order to define the way in which an individual belongs to a given class, we deal with a process that involves three levels, that are:

1. The level of the real-world,
2. The level of the cognitive representation of the real-world

3. The level of textual and graphical "artefacts" or expressions, concerning the real word and its cognitive representations.

The first level is the one of the real-world, which actually is the repository of all our knowledge. It contains facts and things, it represents both the context in which we move our reasoning, and the elements about whom we reason. The second level concerns the cognitive representation of the real world, namely the process we apply to extract concepts from the real world and to assign meanings to them. It means that in the real-world, which holds all our knowledge, we recognize concepts, which have to be represented, and their meanings, that have to be stipulated. The stipulation of meanings occurs in the last level of this representation, in which we define which expressions are useful and appropriate to represent those concepts. The definition of appropriate expressions is achieved through an agreement, which allows us to choose artefacts that can substitute real things. These artefacts are represented by textual and graphical forms, namely words which are used to name and define classes and their characteristics. The process of defining the meaning of a term requests a quasi- Aristotelian approach, in order to establish its telic essence and its relationship with other terms. Applying a quasi-Aristotelian approach means that it is necessary to proceed defining properties of a given class and its individual. In other words, we are led to asking more questions about characteristics of a class and its individuals to describe and classify them. Such description and classification process is established on words and, for this reason, it requires good dictionary definitions, which allow to try and reduce the innate ambiguity of languages. Therefore, it is worth to introduce a precise language, mainly in scientific fields, which aims at creating a one-to-one relationship between terms and specific concepts, reducing ambiguity. Actually, ambiguity is caused by the use of ordinary words in Natural Languages (NLs), which demonstrate vagueness of relationships between terms and concepts. Thus, the attempt of developing a conceptual model of the world requires the use of words provided with specific meanings, in order to avoid vagueness and ambiguity. Thinkers such as Hobbes, Leibiniz and Russell[3], deal with the problem of a mismatching between words and scientific conceptualizations. Special purpose languages have been introduced to overcome the gap between semantic content of common words and the need to represent definite and unequivocal meanings. In such way, we may

[3]In this dissertation, we are not concerned with the vast range of philosophical and scientific developments about this issue. We are just interested in underlying that the problem has been debated by several authors in different times, but it does not seem to be solved.

affirm that each of the three levels in the previous page has its own representative and denotative element which are:

1. Knowledge for the level of the real-world,
2. Concepts for the level of the cognitive representation of the real-world
3. Terms for the level of textual and graphical "artefacts" or expressions.

In Figure 1, we propose a schema for representing elements involved in the processes of identification and definition pertaining the human dimension.

In this schema, we move from a point, with more abstraction, that is knowledge in the real world, to a point with less abstraction, that is terms within the level of textual and graphical "artefacts". Such movement happens thanks to a phase of cognitive representation, in which concepts are elements involved in the above-mentioned processes. Furthermore, such three elements work together as in an iterative process, in which they continuously revise and influence each other by means of a quasi-Aristotelian method.

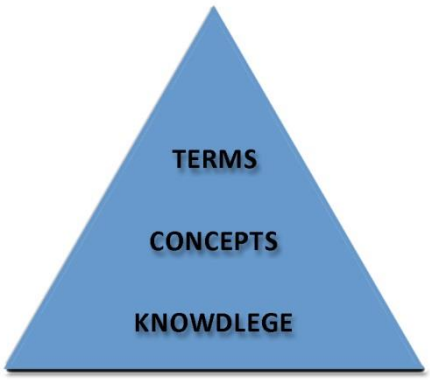


Figure 1.1 – Denotative elements of Human World.

This schema may be compared to Ogden and Richards's triangle, proposed in a study about lexical meaning "The meaning of meaning" (1923), of whom Hanks (2013) proposes a further analysis. Ogden and Richards based their proposal on Saussure's triangle and Pierce's works. Their aim is to demonstrate that there is no full correspondence among word meanings and the objects they (try to) denote. This "relationship is mediated through the conceptual schemes or mental models that language users have in their heads" (Hanks, 2013:330). However, apart from this one, in our work we also suppose the existence of another mediation between knowledge and terms,

represented by concepts. Indeed, this representation can be carried out only by a process of translation in which our knowledge is formalized through the definition of what we want or need to represent using words. The relationship with Sausurre's triangle seems undeniable: knowledge stands for the *signifiant*, concepts represent *signifié* and terminological words are equals to specific *référents.*

As we will see in the following paragraphs, using not ambiguous words in knowledge representation, extraction and management allows the matching of natural languages with machine ones in a more coherent way, guaranteeing higher representative concept levels and improving precision and recall. For this reason, we consider terminology, not generic words, as the expression of a specific knowledge domain[4].

3. Information vs Knowledge

The process, which leads to move from concepts to terms, is founded on the processing of data, in order to define representative elements of our knowledge. In other words, such a process aims at creating a relationship between concepts and terms correlating unstructured data in a structured way. It means that when we associate contexts, meanings and telic intents to data, we may use these data to form, share and re(use) connected information. Thus, discussing about concepts and their representations implies the fact of processing some data, in order to bring together information which may constitute the basis of our knowledge. Indeed, it is worth noting that not all data become information, and that just a subset of the available data may be structured into information. This structuring process entails that we put data into contexts and assign them a purpose to generate meaningful information which may be aggregated as knowledge. However, while data can be fairly easy to define, and also to connect to almost any existing things, Information and Knowledge are difficult to circumscribe and keep separate, being the boundary between them challenging to define precisely. In other words, data are closer to *facts,* which constitute the real-word, than Knowledge and Information, which are the result of reasoning and cognitive processes about these facts. It

[4]For more information, see Paragraph 3.

means that Knowledge and Information belong to a higher level of abstraction and for this reason they are difficult to define and recognize.

In the last years, such difficulty has been often discussed by researchers and scholars, mainly as a key issue in the development of AI systems, but also in the attempt of defining solutions to Big-Data[5] processing. In addition, the need to separate precisely Information from Knowledge shifted inevitably the debate from the philosophic field, which we have been coping with in the previous paragraph, to the CS one.

Indeed, the CS field also deals with the effort devoted to explaining the way in which unstructured data form our abilities to reason and infer applying knowledge. The reason of such an interest concerns the need of defining and identifying both the elements involved in the process and their relationships, in order to represent, manage and extract them adequately.

The most spread attempt to formalize these elements into descriptive process models is the DIKW (Data, Information, Knowledge, Wisdom) Pyramid, also known as 'DIKW Hierarchy'. The model has been credited by Russell Ackoff (1989), although the author did not present the hierarchy as a pyramid. Furthermore, he also interposed an "Understanding" level[6], between Knowledge and Wisdom.

According to Rowley (2007), in DIKW model:

> *Typically information is defined in terms of data, knowledge in terms of information, and wisdom in terms of knowledge, but there is less consensus in the description of the processes that transform elements lower in the hierarchy into those above them*.

Therefore, the basis of such a pyramid is represented by the amount of data on which information lie. In the successive layer, knowledge is presented as a set of information; finally, we find wisdom on the top of the pyramid.

[5]Big data is a broad term which stands for indicating very large and complex data sets which actually represent one of our sources of information and knowledge. The processing of Big Data requires different techniques and methods than the ones used for the other data sets.

[6]Ackoff explains this category as the appreciation of "why" concerning the others elements involved in the process and other questions which we are led to define.

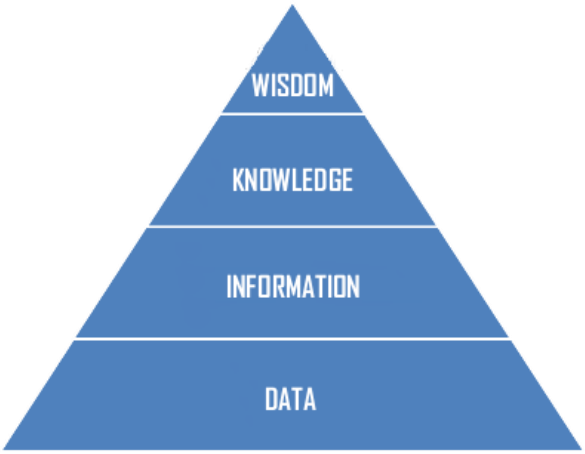


Figure 1.2 – The DIKW Pyramid.

In other words, from unstructured data we may structure information suitable to develop knowledge on which we base wisdom. Therefore, Data refer to symbols we manipulate; Information contain structured data in order to respond to "what", "who", "when", "where" questions. On the other hand, Knowledge represents an application of data and information which allows us to answer to "how" questions; finally, Wisdom represents the evaluated understanding phase (Bellinger *et al.*, 2004). Since the introduction of DIKW Pyramid, several authors propose variations on constituting elements, in order to improve such representation model. Indeed, more than a few authors criticize the DIKW model in its constituents elements, i.e. they do not include data or wisdom, and simply focus on information and knowledge. Among these authors, Frické stresses that the DIKW theory is "essentially conservative over the nature of information" (2009:4). In other words, information holds just observable data and statements which may be inferred from data, while not-observable data and data not inferred from observable data are excluded. Frické assumes that such a positivist approach is traceable in the lacking of a question "why" in Ackoff's list of information seeking questions. The integration of "why" requires that also the other data are included in the Information category, in order to answer to the question. It means that we have to go beyond the observable data in order to receive and use also context information. Indeed, information in context are suitable to establish the reason why we are lead to deal with a specific aspect from the real-world rather than with others.

Another criticized point is the assumption that knowledge is mainly an explicit know-how, which means it is a procedural knowledge, retrievable in instructions and procedures. Such a concept of knowledge involves just data, formalized and structured, and for this reason it fits adequately to the DIKW theory. Nevertheless, interpreting knowledge in such a way entails that it is just a sum of data structured in information, without taking into consideration tacit knowledge. Keeping tacit knowledge out involves a reduction in human contribution to the process related to meaning and reasoning of/about the real-world. We cannot separate data/information from human interpretation without losing key elements involved and necessary to the managing and understanding of real-world representation.

Therefore, the way in which we consider knowledge and information lay the foundation for the subsequent reflections.

We do consider appropriate Gradmann's statement (2010), in which knowledge is defined as information located inside a given context. In his White Paper, devoted to the analysis of KR and Knowledge Extraction (KE) handling in Cultural Heritage (CH), Gradmann refers to the Europeana project[7] as to an attempt of creating knowledge starting from information through semantic contextualization. Indeed, according to Gradmann,

> *Knowledge, then, is information that has been made part of a specific context and is useful in this context. The contextualization processes leading to a specific set of information becoming knowledge can be based on social relations (information as part of a group of people's apprehension of the world, information present in the memory of a person) or semantically based (information related to contextual information via shared properties and thus becoming part of a semantic 'class' of information).*

Therefore, knowledge may be analysed as information inserted in a specific and meaningful context, which means that not-observable data are included in this definition. Contexts in which information are inserted seem

[7]Europeana is a project devoted to the development of Europeana.eu, an internet portal that works as an interface to millions of books, paintings, films, museum objects and archival records that have been digitised throughout Europe (Source: Wikipedia - https://en.wikipedia.org/wiki/Europeana).

strictly related to observable data, and represent constituent elements of knowledge. It means that when we deal with knowledge, we have to consider contextual information, and this is necessary also when we handle such concepts in the CS field, not only in Philosophy.
Furthermore, we share this point of view for two reasons:

1. In our work, we are also aiming at developing a prototype system able to manage CH data and descriptions. Thus, Gradmann's statement and examples may be used to support our approach insofar as they are relevant to our same scientific domain.
2. CH represents a specific domain in which the relationship among data representations, management and extractions has to be particularly ensured, in order to generate and spread knowledge. In fact, semantic contextualization may be used to preserve representation meanings.

Therefore, in the pages which follow we will be referring to knowledge as to a mix which includes information and data, encompassing contextual information. Obviously, such an interpretation influences the way in which we deal with Natural Language Processing (NLP) devoted to Ontology Learning (OL) and Population tasks. For these reasons, we take in account Information Extraction (IE) and KE, considering the representation process suitable for both tasks.

2.1 Information Extraction

For several years, the most general task to develop in IE has been full text understanding, which means that the main goal of research works in this area deals with NLP. A sample of such a purpose are the Message Understanding Conferences (MUC), financed by DARPA (Defense Advanced Research Projects Agency) in order to advance in methods and approaches to IE[8].

[8]"The Message Understanding Conferences were initiated by NOSC to assess and to foster research on the automated analysis of military messages containing textual information.

Although called "conferences", the distinguishing characteristic of the MUCs are not the conferences themselves, but the evaluations to which participants must submit in order to be permitted to attend the conference" (Grishman & Sundheim, 1996).

In this sense, IE could be defined as the process of filtering information from texts, in order to retrieve documents from repositories or also identify relevant entity of a certain class[9], or relations between those entities, and extract relevant arguments in a natural language text[10].

As we will see, during the last decade, the mission of IE has been evolved in more complex goals, which include tasks traditionally related to KE. The reason for such a widening is related to the need of processing increasingly amount of data and information in a consistent way.

Usually, systems devoted to IE aim at analysing information automatically, using a workflow of data manipulation, that is data extracted from an amount of "facts", available in a non-structured form (i.e., newspaper, journal articles, and so on), in order to structure them.

Generally, these systems start with collecting documents, then proceed to transform them in a more readable and analysable way, isolating text fragments. Subsequently, systems extract relevant information from these fragments and finally present the targeted information in a coherent framework.

IE processes are traditionally based on hand-crafted extraction rules or hand-tagged training examples; usually, relations of interest have to be pre-specified by users. These processes show their limits in analysing large and diversified corpora, as those present on the Web. In order to overcome this boundary, IE needs to drop out of relation specifications, which are required during users' queries, and to focus on the identification of all possible relations present in a text.

Focusing on more complex tasks leads IE approaches to improve these systems introducing new extraction paradigms, suitable for the achievement of relation specifications, and so on. A sample of new paradigm for extracting complex information is the Open IE framework.

Open IE is an "extraction paradigm where the system makes a single data-driven pass over its corpus and extracts a large set of relational tuples without requiring any human input" (Banko *et al.*, 2007). Open IE uses a corpus as input in order to extract a set of relations, guaranteeing scalability with the size of the corpus. Thus, Open IE intends to develop extracting ways in which

[9]We define entities of a class any and all individuals belonging to the given class.

[10]In the next pages we will deal with the main tasks of IE and KE.

relationships are expressed in English, not with reference to a specific domain. It does not use lexicalized items, which means that it is based only on syntactic tokens (e.g., part-of-speech tags) and closed word classes (e.g., for, of, in). After the processing phase, Open IE systems extract relational tuples, in the form of Arg1, Pred, Arg2, without relation-specific training data (Etzioni *et al.*, 2011)[11].

Banko *et al.* (2007) suggest that, in these systems, central problems are represented by:

- Incoherent extractions, i.e. lacking of meaningful interpretation in results;
- Uninformative extractions, i.e. omission of critical information.

According to the authors, uninformative extraction is caused by the presence of Light Verb (LV)[12] constructions, which are non-lexicalized items and therefore cannot be processed by the systems.

In order to overcome these issues, the second generation of Open IE introduces the use of generic syntactic and lexical constraints. Such a method is applied in REVERB (Fader *et al.*, 2011), an open extractor, which is developed on a model of verb-based relation phrases[13]. After a phase of matching

[11]As cited in Etzioni *et al.* (2011), samples of Open IE-based systems are TEXTRUNNER (Yates *et al.*, 2007), WOE (Wu and Weld, 2010), and StatSnowBall (Zhu et al., 2009), which apply a procedure structured in three steps: label, learn and extract. TEXTRUNNER, the first Open IE system, is based on a Naive Bayes model, in which training examples are generated from the Penn Treebank . Afterwards, the goal to improve extraction has been sought using a linear-chain Conditional Random Field (CRF) (Banko *et al.*, 2008) or Markov Logic Network (Zhu *et al.*, 2009).

[12]Light verbs are MultiWord Expressions (MWEs), composed by a verb and some additional elements, usually a noun. In these semi-compositional construction, the semantic content is carried by the additional expression (Grefenstette & Teufel, 1995). This kind of MWEs, also called operators, operator verbs, complex predicates, etc., have been investigated by different linguistic schools. Being Lexicon-Grammar our theoretical and practical framework (see Chap. V), we will refer to these verbs as Support Verbs and Support-Verb Constructions (Gross, 1986b), and we will use these labels to identify such a notion.

[13]It means that "the system takes a sentence as an input, identifies a candidate pair of NP arguments (arg1, arg2) from the sentence, and then uses the learned extractor

between noun phrase (NP) arguments and relation phrases, REVERB assigns a confidence score to the results obtained, applying a logistic regression[14] classifier, trained on Web sentences, presenting shallow syntactic features[15].

A similar system presents several boundaries. The first one – a boundary the authors intend to cope with in their future works – is the fact that not all relationships are binary. Indeed, various verbs may take three or four arguments, as for instance *to move* in the sentence "**Max** moves the **chair** from the **dining room** to the **kitchen**". For this reason, the logistic regression model, which is based on binary relations, may not be highly accurate.

The second one is related to the presence of some important relationships which may not be expressed by verbs or verb phrases (VP). For instance, some relation phrases are expressed by a combination of a verb with a noun, namely a LV construction, in which the noun carries the semantic content of the predicate. Thus, Fader *et al.* (2011) propose the use of syntactic constraints to included nouns in relation phrases during the extraction process. This means that, in order to overcome these boundaries, it is worth integrating semantic values, namely using a fine-grained analysis.

Such an integration of semantic analysis has been also used to deal with synonymy, polysemy, and similarity of "relation phrases"[16].

Indeed, since the introduction of the Open IE paradigm, several approaches aimed to integrate semantic analysis within relation extraction

---

to label each word between the two arguments as part of the relation phrase or not" (Fader *et al.*, 2011).

[14]This is a direct probability model, developed by D.R. Cox (1958 and 1958b). It is used to predict a binary response based on one or more predictor variables. By estimating probabilities, it measures the relationship between the categorical dependent variable and one or more independent variables, which are usually (but not necessarily) continuous. See also http://en.wikipedia.org/wiki/Logistic_regression.

[15]Shallow syntactic features refer to text chunking results, namely a technique for recognizing simple sentence structures.

[16]In English, a phrase is a small word group, that is an immediate constituent of a sentence (a sentence may include several phrases. Phrases cannot include sentences), due to the fact that a phrase (every kind of phrase: noun, adjective, verb, adverb and prepositional ones) does not mean a complete idea, lacking of a subject, a verb and a predicate (in following chapters, we will show as this sentence structures is strictly related to a Resource Description Framework – RDF – triples). In our dissertation, we always use the word 'sentence' to indicate a complete word sequence in which we can recognize a structure expressing a thought.

processes, trying to ontologize semantic relations. The goal of any ontologization process is to describe semantic relations by means of ontological constraints.

Ontologization process in IE paradigms also founds the introduction of models devoted to describe distributional semantics[17], attempting to overcome the previous boundaries.

For example, Moro & Navigli, (2013) try to achieve semantic integration into the Open IE paradigm basing it on deep syntactic analysis and distributional semantics, by means of a shortest path kernel and soft clustering. Actually, distributional-semantic approaches bring into references to a semantic space in which it is possible to evaluate semantic similarity between two words. The semantic content of a given word is defined by a vector, which is placed inside a Cartesian coordinate system. Such space stands for the linguistic context in which a word may occur. Among the various models of distributional semantics, the most widespread are Latent Semantic Analysis (LSA), Hyperspace Analogue to Language (HAL), and lately, Random Indexing. The main difference among these semantic distributional approaches is represented by the definition they assign to the semantic space or by the elements which may constitute the context base.

Another difference is retrievable into methods applied to describe semantic similarity among words. Usually, stochastic methods are used to describe such similarity through statically distribution of word co-occurrences in texts or corpora[18].

The main boundary of these approaches is the lacking of references to Structural Linguistics. In fact, even if semantic representation is related to the behaviour of arguments co-occurring in specific sentence contexts, such approaches do not either apply or preview an accurately formalized linguistic description. In other words, distributional hypothesis[19] of a given word and its

[17]"Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical representation (e.g. their string format)".

Source: https://en.wikipedia.org/wiki/Semantic_similarity.

[18]For more information, see Chapter III.

[19]As we will see, the foundation of distributional hypothesis are traceable in Z.S. Harris' work (1968).

predicate-argument structure are just dealt with by means of clustering techniques and kernel methods.

Indeed, the extraction task should provide for the identification of syntactic and semantic properties of a word in relation with other words according to the operator-argument structure of a sentence. Meanings have to be described on the basis of operators with reference to the arguments they select (Harris, 1976), in order to interlink the syntactic and semantic components of a language. This is the notion of 'semantic role', developed by different linguistic theories. Generative linguistics considers semantic roles as the deep structure of a language, that is how we organize concepts and establish relations among them. On the other hand, the surface structure is constituted by the representation of this organization in the grammatical form of a language (Marano, 2012).

Anyway, the attempt of (re)constructing such relations among words introducing ontologies in IE techniques leads to develop new methods, which involve mainly knowledge. It means that when we want to introduce ontological semantic analysis, we have to deal with tasks concerning knowledge and its representations and extraction.

2.2 Knowledge Extraction

By reason of previous motivations, in the IE field researches and scholars aim at integrating knowledge into the process of textual analysis. Inside texts, such an integration is suitable to recognize entities, namely primitives, and relations among them. In this regard, we may define Knowledge Extraction (KE) as an attempt to retrieve ontological relations, namely to (re)create knowledge from different textual sources (i.e. structured, unstructured and semi-structured).

Indeed, even if KE faces many of IE issues, basically it attempts to deduce a rule base or a domain model on the basis of technical texts (Cowie & Lehnert, 1996); such attempts include a strong machine-learning component, in addition to the NLP one (Matwin & Szpakowicz, 1993).

As we have previously stated, KE is the retrieval from structured and unstructured sources of text elements and data bringing and representing knowledge. This brings to the consideration that extracting semantic content is possible only in presence of concept-based systems using sets of features

which must be (pre)defined in order to represent conceptualizations and their formalizations.

KE from texts has become a key semantic technological procedure. Initially, in ontology learning studies, KE had not a relevant diffusion, due to the fact that manual ontology design was the practice mainly used. The scenario started to change when researches begun to use Web resources in order to populate KBs by means of structured and unstructured contents (Gangemi, 2013).

Recently, there exist deep KE techniques, based on hybrid approaches which combine trained models and rule-based methods, mainly thanks to the using of existing knowledge coming from Linked Open Data (LOD)20. For instance, some machine learning techniques aim at inducting rules from sets of examples[21].

The process of extracting useful knowledge from large-scale text collections, derived from the Web, takes advantage of text and data mining techniques. In particular, data mining refers to the process of defining useful KE rules and patterns analysing data. In the last years, various approaches have been proposed for developing faster algorithms, which allow processing the growing volumes of data and to find meaningful patterns. However, machine-learning techniques do not seem to be advanced enough to process and extract knowledge from large-scale repositories and yet reporting a high score of precision and recall. Extracted knowledge, as a NLP result, is the basis on which ontology population relies. In order to obtain an efficient KE, it is necessary that an annotation process preserves links between concepts and machine formalisms. In other words, the aim is to save the relations between formalized knowledge and its linguistic representation, providing some data about other data, that is describing resources in order to keep meaning unchanged.

[20]In the following chapters, we will introduce specifically Linguistic Linked Open Data (LLOD), which are closer to our thesis aims. Here, we only cite the four principles of Linked Data, stated by Tim Berners-Lee (2006):

1. Use URIs as names for things. [URI = Uniform Resource Identifier]
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URIs so that they can discover more things.

[21]For more information, see Chapter 3.

Resources may accurately be described by means of annotation. The definition of annotation may indicate both the process and the result of annotating (Handschuh, 2005). We can recognize three kinds of annotation: informal, formal and ontological (Oren *et al.*, 2006). The first one is not formalized, i.e. it looks more like a jotting down on a side of a book page. Instead, formal and ontological annotations use formal languages and, as such, they are machine-readable. The difference between these last two is that in order to represent concepts, ontological annotations use only ontological terms. Indeed, in ontological annotations terminology has a commonly understood meaning that corresponds to a shared conceptualization called ontology (Gruber, 1993).

For these reasons, recently a new KE task has been identified, that is Ontology-based Information Extraction (OBIE)[22]. OBIE uses ontologies and their specifications to "drive" the information extraction process. During annotation, terms and concepts in the source ontology form the basis for term matching.

Applying ontological annotations to resources may guarantee semantic disambiguation and at a higher representative level, due to the fact that ontologies always refer to a specific domain of knowledge. As we have demonstrated for Aristotelian categories, it seems not possible to create a general classification of the world. Indeed, during the specification process, our answers about individuals and their properties may define only a particular segment of the whole existing knowledge.

From this statement, we may derive some questions to manage our discussion. Is it possible to overcome the ambiguity and vagueness of common words? Is it possible to annotate semantically common words without referring to their context and domain uses? How can we evaluate both the semantic value and the meaning without evaluate the lexical, syntactic and semantic characteristics of words in relation with other words?

4. Terms as Conceptual Referents of a Specific Domain

The purpose of this paragraph is not to bring further contributions to the already existing discussion about lexicography and terminology, be it in terms

[22]For more information see Chapter II.

of differences, similarities, inclusion and exclusion[23]. We just want to further specify why terms are the constituent elements of knowledge and which is their role in KP tasks. Terminology copes with concepts and their denominations in special subject fields, which means that it may be defined as being concept-based[24]. In other words, terminology relates to the relationship existing between concepts and their referents. These referents comprise words and phrase and also symbols, drawings, formulae, and so on, that is they do not have to be necessarily classic lemmatized lexical entities/entries. For its characteristics, terminology is strictly interconnected to specialist knowledge and special languages. For this reason, KR is heavily based on the use of terminology, due to the fact that many terms have precise meanings in a specific domain but not in others.

On the contrary, lexicography is engaged in at recording words of the general vocabulary of a language into specific formalized format (e.g., alphabetically), and merely providing supplied information (spelling, pronunciation, grammatical class, etc.). Therefore, lexicography investigate everyday language, taken out of a specific domain, and, as such, endowed with a high level of abstraction. Even if it describes use contexts of lemmatized words, a not terminological dictionary lists (in a neutral way) simple words, not MultiWord Expressions (MWEs)[25].

---

[23]The opposition could be traced with the introduction of the term *terminography*, in 1975. This ISO 1087 standard aims at replacing the terms *terminological lexicography* and *special lexicography* (cf. Bergenholtz & Tarp 1995:10; Humbley 1997:14).

[24]Philosophical foundations of terminological studies are initially traceable during the 17th and the 18th century. Indeed, Wolff deals with the evolution of German as a 'language of science'; Leibniz proposes an ideal language of science; while Kant develops a constructionist concept theory. Afterwards, in 19th century, Bolzano, Hartmann and Brentano cope with terminology, until Neo-Aristotelian Epistemology, namely virtue epistemology that stresses the importance of intellectual (epistemic) virtues. Among various works on epistemology we have to cite Dewey and Pierce and Eco' semiotic studies.

[25]We define this recording as a neutral action, even if, in lemma description, general dictionaries often report references to the specific domain(s) in which the word is used. This happens mainly for those specialized words that are largely diffused in common language. Therefore, their specific meanings overcome the boundaries of domain languages, becoming part of everyday language use. Anyway, this process happens only for simple words, which are lemmas with a highest level of ambiguity. Compound words, MWEs, Multi Word Units (MWUs) are excluded from common dictionaries. In Chap. V, we

The main difference between terminology and general terms is that the representational expressions used by the former are in a one-to-one correspondence with the entities/concepts in a given domain. Therefore, we may describe a specific domain of knowledge and its concepts through terminology, not through lexicography. Actually, terminology allows us to choose those words and relationships between words which are able to represent specific concepts univocally in a given field. In brief, we proceed in an Aristotelian way, in order to response to more specific questions about what we want to describe.

Consequently, a coherent and consistent linguistic formal description is crucial and indispensable to achieve a correct semantic representation of a specific knowledge domain (di Buono, 2015). As we have seen, terminology presents two levels: the first one is essentially linguistic-based, the second level is a conceptual one. The linguistic level complies with language for special purposes whereas the conceptual level is related to concepts in a given domain knowledge. Due to these characteristics of terms, in terminological study evolution, different knowledge management, ordering and retrieving systems have been applied. Among these systems, which present their own data models, purposes and traditions, we may find classification systems, thesauri, indexing systems, taxonomies nomenclatures, and mainly ontologies. This means that we may use ontologies and their data model also in order to describe relationships among terms, which namely are concepts.

Considerations, which follow, spring from the idea that the real-world and the machine one use two different representations, which means different denotative elements.

In Figure 1, we propose a model of identification and definition of the real world which holds knowledge, concepts and terms. Taking up Figure 1, in which we present the scheme for the real word, we create here a correspondence between real world and machine world (Figure 3). Thus, we assume that the Human World (HW) is mirrored by the Machine World (MW), which on its turn holds three denotative elements suitable for concept, identification and definition.

---

will introduce the concept of Atomic Linguistic Units, that in our opinion may be used to define the whole set of not simple/complex words.

These MW denotative elements are data, metadata and NLP Formal Languages (FLs). Thus, Knowledge is related to the quantity of meaningful data we may process, Concepts correspond to metadata[26] and Terms match with NLP FLs. In order to allow the processing of a representation through a computer, it is necessary to use a formal semantics description, converting it into a machine-readable formal representation. The choice of which formal language has to be used depends on the complexity of what we want to express and on the kinds of reasoning we want to apply.

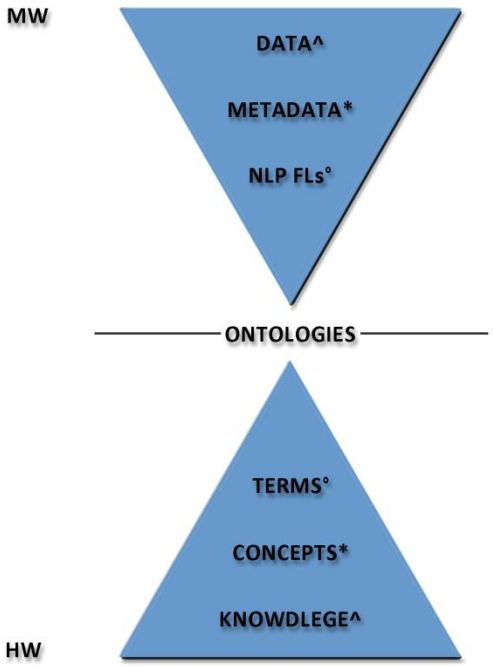


Figure 1.3 – Human World and Machine World constitutive elements.

In this model, we may identify ontologies as the trigger between HW and MW representations. Indeed, due to the fact that they focus on terms meaning and on the nature and structure of a given domain, ontologies are suitable to match human and machine representations. In order to justify our standpoint, firstly we have to introduce metadata schemata and draw upon the theory of

[26]Usually, we distinguish metadata in structural and descriptive metadata. Structural metadata refer to the structure of data. On the other hand, descriptive metadata are individual instances of application data or the data content.
Source: https://en.wikipedia.org/wiki/Metadata.

formal languages. Indeed, such approach may efficiently be used to explain formal representations, as regards both metadata and conceptual models.
Therefore, during KP, that is during representation, retrieval and extraction steps, we always have to keep in mind our attempt to create a trigger between HW and MW.

5. Standard Metadata Schemata

In this paragraph, we present an introduction to standard metadata schemata, to underline those aspects useful to prove and/or support our considerations. Hence, we do not want to provide here a complete guide about this topic. Among standard metadata schemata, we chose to deal only with the main ones, and more specifically with those on which are based the conceptual models used in CH domain.

Resource Description Framework

The Resource Description Framework (RDF) provides a common data abstraction and a syntax for all Web content. The RDF Vocabulary Description language (RDFS) and the Web Ontology language (OWL) together provide a common data modelling (schema) language for data in the Web. The SPARQL Query Language and Protocol provide a standard means for interacting with data in the Web.

Dublin Core

The Dublin Core Metadata Initiative[27] has been developed since 1995 by the Online Computer Library Center (OCLC) and the National Center for Supercomputing Applications (NCSA). Its goals are to create standards for the description of online resources, to combine metadata vocabularies of different metadata standards, and to guarantee interoperability for metadata vocabularies in the Linked Data cloud. The Dublin Core Metadata Element Sets is composed by 15 elements plus their general definitions, finalized in December 1996. Such elements may be associated with a controlled vocabulary, in order to promote global interoperability. “In the element descriptions, each element has a descriptive name intended to convey a

[27] http://dublincore.org/.

common semantic understanding of the element, as well as a formal single-word label intended to make the syntactic specification of elements simpler for encoding schemes" (Weibel *et al.,* 1998).

SKOS

Simple Knowledge Organization System (SKOS)[28] develops specifications and standards to support the use of Knowledge Organization Systems (KOS), as thesauri, classification schemes, subject heading systems and taxonomies[29], using RDF[30].

KOS, also known as "controlled structured vocabularies", has been developed to organize large collections of CH objects, to be used in both modern and traditional information systems.

SKOS Specifications are published as a W3C Recommendations. In this data model, "concepts can be identified using Uniform Resource Identifiers (URIs), labelled with lexical string, assigned lexical codes, linked to other concepts and organized into informal hierarchies and association networks, aggregated into concept schemes, grouped into labelled and/or ordered collections, and mapped to concepts in other schemes"[31].

6. Formal Languages for NLP

This section concerns the classical formal language theory, based on Chomsky's investigations about natural language. However, we choose to overlook much of automata constructs and computability issues. In fact, our aim is to introduce only some of the principles and theoretical tools that will be resumed in a more detailed consideration in the next chapters.

In CS, formal languages are normally defined as an alphabet, formed by a set of symbols, and by the rules useful to produce formal expressions using by

[28]http://www.w3.org/2004/02/skos/intro.

[29]There is not an absolute distinction between thesauri and classification schemes or taxonomies, although some properties can be used to broadly characterize these different families. Source: BS8723 Structured Vocabularies for Information Retrieval Part 3: Vocabularies Other Than Thesauri, British Standards Institution (BSI), 2005.

[30]RDF provides a common data abstraction and syntax for the Web.

[31] Source: https://www.w3.org/TR/skos-reference/.

means of the alphabet. Formal Language Theory considers a language as a mathematical object. In this way, we can define language as a (presumably infinite) set of strings, admitted by the language itself, over a finite alphabet.
In linguistic studies, formal languages are used to analyse human languages, mainly preferring a generative approach[32]. Such approach defines a set of rules to express a grammar, on which any sentence of a specific language may be constructed. Therefore, a grammar describes sentence forms, not their meaning[33].

According to Chomsky's Syntactic Structures (1957), knowledge of language is modelled by means of a formal grammar. Indeed, formal grammars account for how we are able to produce and process an infinite number of sentences, using a finite set of grammatical rules and terms.

In his hierarchy (also referred to as Chomsky-Schützenberger hierarchy), Chomsky proposes different kinds of formal grammars, together with their related languages. Each class is characterized by an expressive power, which increases by the level and generates a wider formal language.

Grammars that fall in rewriting systems are defined as a finite state of rules, which are able to generate language systems. The language generated by a grammar is a set of terminal symbols that can be derived from the starting symbol, using production rules of the grammar itself.

Formal grammars are defined as a finite set of production rules, formed by tuples. In such tuples, from a starting nonterminal symbol on the left (S), the right side may be composed by a set of terminal symbols or a set of nonterminal symbols.

Therefore, we may have tuples as follows:

$S \rightarrow ABC$

[32]As we will see, this means that by means of such grammars we may define rewrite rules of phrase structure (Chomsky, 1957). For more information, see Chapter IV.

[33]In Chapter IV we will deal with some of these grammar formalisms. However, we just present the ones which are applied most frequently in Ontology Learning and Population. Thus, we do not deal with Linear Indexed Grammars (LIG) or Xerox Finite State Tool (XFST), which is a general-purpose utility for computing with finite-state networks, suitable for morphological analysis. However we will introduce other similar formalisms, i.e., Tree Adjoining Grammars, Combinatory Categorial Grammars and so on, which are all based on contextual grammars (see afterward).

S → abc
S → ε (where ε indicates an empty string).

According to Chomsky, formal grammars may be of four different kinds:

0. Unrestricted grammars which include all formal grammars
1. Context-sensitive grammars which generate context-free languages
2. Context-free grammars useful to produce context-free languages
3. Regular grammars which generate regular languages.

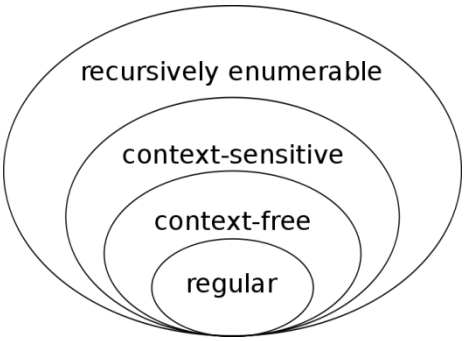


Figure 1.4 - Chomsky-Schützenberger hierarchy[34].

Each class of grammars and generated languages presents different production rules and restrictions. In addition, it may be recognized through different automata. In order to define a language, we may use three methods: regular expressions, automata and grammars.

An automaton is an abstract state machine based on a recognition approach. Stephen Kleene has introduced automata in the '50s. His aim is to substantiate the correspondence between such model and the description of symbol sequences. In order to obtain this equivalence, he uses only three logical primitives: set union, set product and iteration.

Kleene's automaton is equipped with a finite memory, which represents a change with respect to Alan Turing's proposal, based on an abstract state machine with unbounded memory. Such memory is accessible through a stack, which is a restricted mode. In this way, Turing lays the groundwork for pushdown automata. The Turing Machine (Turing, 1936 and 1939) manipulates

[34]Image taken from https://en.wikipedia.org/wiki/Chomsky_hierarchy.

symbols applying a table of rules, and it is equipped with a non-finite memory formed by a strip of tape.

Each class of Chomsky's hierarchy corresponds to, and is recognized by, a specific type of automaton. Due to the equivalence between grammars and automata, formal language theory can be analysed through either a generative approach or a recognition one.

After presenting the generative approach in this first part of the paragraph, we will now proceed to present the recognition one. According to this approach, a language is formed by a set of strings which belong to that language and therefore are accepted by a given automaton. The recognition process starts from an initial state and transits through different states, led by symbol strings. The process ends when the whole string is read and the state can be rejected or accepted.

On the basis of the proposed approach, we can systematize Chomsky's hierarchy associating grammar classes to the respective automata and languages.

Table 1.1- Language classification based on Chomsky's Hierarchy.

Chomsky	Grammar	Automaton	Language
Type 0	Unrestricted	Turing	L_0 – Recursive enumerable
Type 1	Context-sensitive	Linear-bounded	L_1 – Context-sensitive
Type 2	Context-free	pushdown	L_2 – Context-free
Type 3	r./l. linear	Finite-state	L_3 – Regular

L_0 are composed by sets in which elements can be enumerated through an algorithm. In L_1 monotone and context-dependent grammars have the same expressiveness. Context-free grammars are the syntactic basis for most of programming languages. Finally, regular grammars may be used to define lexical structures of programming languages.

In this essay, we will mainly focus on Finite-State Automata (FSA), which are a part of the Linguistic Resources (LRs) we will be presenting in Chapter V.
There are two kinds of FSA: deterministic or non-deterministic.

A deterministic FSA is a machine which takes a sequence of symbols as input, giving a positive or negative response as output, according to the recognition or the rejection of the sequence. Automaton states represent its memory. FSA moves from state to state, while it reads the sequence of

symbols. An FSA is said to be deterministic when given a state and an input symbol, it has only one way to proceed from the initial to the end state.

Instead, a non-deterministic automaton may choose different paths to move from the initial to each one of the following states. While choosing a specific path, the automaton rejects all other possible paths. In this way, non-deterministic automata express the possibility of proceeding in more than one determinate way to reach the final state.

When an FSA can choose among two or more paths, non-determinism can be interpreted in three ways:

1. The automaton knows the right path. This is called the Oracle way.
2. The automaton clones itself trying every paths. This is Parallelism.
3. The automaton build a backtracking way, in which it saves its status (as state and position in the input symbols) and tries one of the paths.

While automata are easy to implement, they may be hard to create and understand. Starting from the idea that regular languages are similar as for features like union, concatenation, and so on, we may define both an algebraic language, in order to specify regular languages, and a way to convert it to/from automata. This process generates and is based on regular expressions.

Pushdown automata are composed by a non-deterministic FSA with ε transitions and a stack of unlimited size. Both current symbol in the input string and the topmost symbol in the stack can change state. A pushdown automaton is a tuple, in which there are:

1. Finite sets of states
2. An input alphabet
3. A stack alphabet
4. A transition function
5. An initial state
6. An initial symbol in the stack
7. Sets of final states (Aldini, 2014).

These formal models have many applications in fields other than natural language studies; for example, recent advances in natural language processing are spurred by them. Formal languages and grammars also define families of

formal languages, which are applied in many computer science applications. Actually, context-free languages are most widely used to describe the syntax of programming languages.

7. Formal Models for Archaeological Data

Archaeology represents a wide knowledge domain with differing levels of significance and contents. This comes from the presence of various assets; the term Archaeological assets[35] covers a whole range of remains: objects, ancient buildings and archaeological sites. Among these assets, we may distinguish two group: below ground remains and above ground remains.
The cataloguing process of both assets amounts to actions of registration, description and classification of those assets.

In the CH documentation, that is more generally speaking in the Archaeological one, source of information are formed by free text and structured metadata records. Usually metadata are basic and standardized, while free text provides detailed descriptions and additional information. Being formalized, metadata are not difficult to be processed by machines. This is not the case with free text contents, which represent a more relevant source of knowledge about assets and their specific domain. In this sense, the fundamental challenge is making CH data, information and knowledge available and sharable working on the semantic integration of the heterogeneous schemata used by all different content providers. In fact, "it is a common opinion that the diversity and epistemological richness of cultural heritage provides an excellent field for the deployment and experimentation of Semantic Web-based systems" (Bordoni, et al., 2013).

The representation model of archaeological data is an effortful task, due to the complex nature of the domain. Before the introduction of machine readable formats, in CH documentation classification standards have been developed for data sharing, in order to improve content management and reduce cataloguer efforts. Domain concepts and their relationships have been proposed in the form of classification systems, list of terms, catalogues,

[35]The English National Planning Policy Framework (NPPF) defines 'Heritage Asset' as 'A building, monument, site, place, area or landscape identified as having a degree of significance meriting consideration in planning decisions, because of its heritage interest'.

thesauri and more general schemata. In this way, CH domain has worked on terms standardization and common classification standards, providing effort for management, preservation and archival of its assets. The problem with CH object representation defines a field of interdisciplinary research and it is also the area in which notable results of general value have been obtained.

Because of the need for information integration, different metadata schemata have been developed into national and international frameworks, to guarantee data interoperability. This is the case with the LIDO and CARARE metadata schemata.

In the following pages, we will introduce the main standards and conceptual models developed for preserving semantic interoperability and mapping metadata in CH domain.

The CIDOC Conceptual Reference Model

Developed since 2006, the CIDOC Conceptual Reference Model (CRM)[36] aims at providing semantic definitions to describe implicit and explicit concepts and relations between CH objects and museum documentations. It is a formal ontology, which allows integration, mediation and interchange of heterogeneous information. CIDOC CRM only defines basic semantics for database schemes and document structures. It does not offer an overview of the terminology used in these structures, and it represents mainly a guide for conceptual modelling, usable as formal language, supporting algorithms of automatic data transformations without loss of meaning. As stated in (Doerr, 2003), this object-oriented semantic model is composed of 90 classes (which includes sub-classes and super-classes), 149 unique properties (and sub-properties)[37], and it is compatible with RDF.

Europeana Data Model (EDM)

Europeana Data Model (EDM) proposes a structure for the data ingested, managed and published by Europeana. It is an improvement of Europeana

[36]A detailed analysis of this conceptual model, together with its property and classes, is provided in Chap. V.

[37]In Par. 5.4 we will discuss specifically of how these classes and properties have been used in order to accomplish our thesis goals.

Semantic Elements (ESE), the basic data model of the EU project[38]. Aiming at being an integration of different CH contents, this model intends to offer a way to integrate any element present in providers' description.

Lightweight Information Description Objects (LIDO)

As a result of the joined work of CDWA Lite, Musemudat, SPECTRUM and CIDOC Conceptual Reference Model (CRM) communities, LIDO is a schema used for encoding core records concerning any kind of cultural heritage objects.

Indeed, LIDO39 aims at integrating information provided by organizations in different metadata formats, supporting a full range of descriptive information about museum objects. It defines 14 groups of information, made up of nested set of 'wrapper' and 'set' elements. The concept of events is borrowed from CIDOC CRM, in order to represent in a consistent way all entities involved in an event. Therefore, the creation, collection and use of an object may be described as events, associated to entities as actors, places and dates. "LIDO also allows the recording of information about the sources for data (e.g., in a book) and controlled terminology (e.g., the identification code for a term in a thesaurus)".

In LIDO, information are conceptually organized in seven areas, divided in descriptive and administrative kinds. The first ones concern object Classification (information about the type of object), Identification (basic information about the object), Events (events in which object has taken part in) and Relations (relations of the object to). Administrative information refer to Rights Work (information about the rights associated with the object), Record (basic information about the record) and Resource (information about digital resource being supplied to the service environment) (Coburn *et al.*, 2010).

CARARE

CARARE[40] 2.0 "takes into account the experience gained from mapping more than 40 datasets from 20 different countries to version 1.0 of the CARARE

[38]http://www.europeana.eu/portal/.

[39]http://www.lido-schema.org.

[40]http://www.carare.eu/swe.

metadata schema during the CARARE project, and from transforming CARARE metadata to EDM to contribute to Europeana".

"The schema is an application profile based on MIDAS Heritage and the CIDOC CRM. MIDAS Heritage is a detailed standard intended for the full documentation of all aspects of heritage management, not all of which are relevant to the CARARE service environment. The CARARE schema's focus is on the detailed description of heritage assets, events in which the asset has been involved and digital resources which are available online and their provenance. It follows the structure of MIDAS Heritage enhanced by the expressiveness of LIDO and EDM" (Fernie *et al.*, 2013).

II - ONTOLOGIES AND KNOWLEDGE PROCESSING

The task of classifying all the words of language, or what's the same thing, all the ideas that seek expression, is the most stupendous of logical tasks. Anybody but the most accomplished logician must break down in it utterly; and even for the strongest man, it is the severest possible tax on the logical equipment and faculty.
Charles Sanders Peirce, letter to Editor B. E. Smith of the *Century Dictionary*

1. Different Types of Ontology

Processing knowledge through ontologies seems to guarantee more accuracy than other representation/classification/extraction methods. However, ontologies require continuous updates, due to the main characteristics of knowledge, which as stated elsewhere is fluid and is not composed by a finite set of information. Therefore, ontologies have to be continuously updated if we want to ensure an accurate representation of a domain knowledge.

In this chapter, before dealing with the main tasks related to ontological KP, we want to recall the definition of ontology, together with its structure and goals.

Thus, we propose again Gruber's description (1993):

> *An ontology is a formal, explicit specification of a shared conceptualization. 'Conceptualization' refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena. 'Explicit' means that the type of concepts used, and the constraints on their use are explicitly defined. 'Formal' refers to the fact that the ontology should be machine readable. 'Shared' reflects that ontology should capture consensual knowledge accepted by the communities.*

By the time, the term ontology together with Gruber's definition, have become widespread.

In spite of this, we think it necessary to further specify the main aspects deriving from Gruber's statement, that is to say: how a conceptualization can be formalized and also, how it can be formally represented.

Generally speaking, the adoption of given formal characteristics comes directly from the use of specific formal systems. This means that also an ontology can be singularised by the application of a distinct formal language, which may be structured and composed by:

- A set of symbols;
- A grammar, which specifies rules for well-expressed formulas;
- Finally, a set of axioms and inference rules for reasoning over this language.

We have already conveyed the issue of knowledge representation as a problem related to transmitting concepts and meaning. Therefore, and as a consequence to this, we may now state that an ontology represents the attempt to overcome the boundary between human and machine semantics. To achieve this task, ontologies present conceptual levels, both intentional and extensional (Guarino, 1998). The extensional level is related to domain instances, while the intentional level describes a domain of specific knowledge. Besides, we may have two more types of ontologies, which are:

1. Upper ontologies;
2. Domain ontologies.

Upper ontologies, also called foundation ontologies, aim at describing general entities and contain generic specifications, which means that they are not domain specific. Thus, upper ontologies are the expression of a generic real-word knowledge.

Three of the most spread foundation ontologies are Cyc (Lenat, 1995), DOLCE and the Suggested Upper Merged Ontology (SUMO) (Pease and Niles, 2002).

Cyc aims at formally representing facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life[1].

DOLCE, which stands for Descriptive Ontology for Linguistic and Cognitive Engineering), "has a clear cognitive bias, in the sense that it aims at capturing the ontological categories underlying natural language and human common-sense" (Mascardi *et al.,* 2006).

SUMO results from the merging of different upper-level ontologies. Due to the generic conceptualizations represented, learning tasks for upper ontologies seem impossible to structure and exploit, also because the knowledge such ontologies express is not explicitly lexicalized inside texts.

On the other hand, domain ontologies are more specialized, and depict a specific domain knowledge. Within such ontologies, entities and relationships may be often easily recognized and extracted, due to the fact that they are expressed directly in texts. As we have seen in describing how terminology is used, a domain ontology is also characterized by a minor influence of word sense ambiguity.

A domain ontology may be represented by a tuple composed of <C, H, R, A, I>, in which:

- *C represents the set of classes. E.g., Animal, Human, etc.*
- *H represents the set of hierarchical links between the concepts. E.g., is-a (Feline, Animal).*
- *R, the set of conceptual links. E.g., eat (Herbivores, Plant)*
- *A, the set of axioms, i.e., the rules that govern this domain and make a reasoner able to infer new information.*
- *I the set of instances, i.e., the objects of the world which can be categorized into the ontological classes C* (Zouaq, 2011).

According to Sowa[2], on the basis of how they define their categories, we may also distinguish three types of ontologies:

1. Formal ontologies

[1]The Cyc project was founded in 1984 by D. Leant as a lead project in the Microelectronics and Computer Technology Corporation (MCC). http://www.cyc.com/.

[2]http://www.jfsowa.com/ontology/gloss.htm last edit 2001.

2. Terminological ontologies
3. Prototype-based ontologies.

Formal ontologies are "a conceptualization whose categories are distinguished by axioms and definitions. They are stated in logic that can support complex inferences and computations." (Biemann, 2005).

Terminological ontologies do not require a full specification provided by axioms and definitions. For instance, WordNet[3] is the spread example of terminological ontology. It does not define completely concepts, because their positions, with respect to one other, are only partially determined. In fact, WordNet just specifies categories through relationship of subtype-supertype and type-whole.

On the contrary, and even if it is based on terminology, a prototype-based ontology differs from a terminological one, due to the fact that its categories are defined by typical instances, or *prototypes*, which are preferred to axioms and definitions. In prototype-based ontology, the semantic distance among entities in specific categories is measured on the basis of a prototype, that is an example. For this reason, in order to measure semantic distances, often methods have been applied that learn from examples (e.g., neural networks, cluster analysis and statistics[4]) (Sowa, 2001).

Semantic distance, or semantic similarity, is related to the measurement of similarity degree "between concepts/terms included in knowledge sources, in order to perform estimations" (Slimani, 2013).In other words, computing the semantic distance between concepts/terms allow us to identify concepts which have some 'characteristics' in common. Relationships among concepts seem hard to be formally defined by humans; despite of this, estimating relatedness among them appears easier than defining it. Indeed, we can recognize the degree of semantic similarity between 'apple' and 'peach' as a stronger relation than the one existing between 'apple' and 'tomato'. Nevertheless, if our knowledge is applied in the Informatics domain, we can also recognize a relation between 'apple' and 'laptop'. Lacking a formal definition of this relatedness among concepts and terms, solving word sense ambiguity, and

[3]For more information, see Chap. IV.

[4]For more information, see Chap. III.

consequently, retrieving and extracting correct knowledge, seems an unachievable task.

Because of this, semantic similarity measure is always performed on the basis of a given ontology. Indeed, ontologies allow the use of structured knowledge representations, able to provide a semantic representation of terms. The concepts of each knowledge source have to be represented by specific domain ontologies, if we want to recover the similarity between concepts/terms.

"The similarity between concepts or entities can be identified if they share common attributes or if they are linked to other semantically related entities in an ontology" (Slimani, 2013). In other words, applying an ontology seems to ensure an improvement of similarity measure among concepts, due to the fact that we may compare common attributes.

Actually, most applications of intelligent knowledge-based and semantic information retrieval systems take advantage of semantic similarity calculation, which may be computed by means of various methods, e.g., statistical, stochastic and rule-based.

2. A Survey on Ontology Learning and Population

Ontology Learning (OL) represents the process through which we extract conceptual knowledge and elements from different inputs, in order to build an ontology. Several methods are applied in OL process, such as Machine Learning (ML), Knowledge Acquisition (KA), NLP, Information Retrieval (IR), AI, reasoning and database management.

In order to extract concepts from texts and provide inferences on ontological knowledge, most researches are focused on different learning processes for populating ontology. Supporting the construction of ontologies and populating them with instantiations of both concepts and relations is commonly referred to as onto OL (Lehmann & Völker, 2014).

In 2006, Cimiano proposes an OL layer cake, in which the author organizes various OL tasks in a layer diagram, in order to present conceptual dependencies among such tasks. Indeed, as showed in Figure 1, outputs in a lower layer typically become inputs in the higher layer. Thus, "for example, in order to extract relations between concepts, we should consider the

underlying hierarchy to identify the right level of generalization for the domain and range of the relation" (Cimiano *et al.,* 2009). Layers of Terms and (Multilingual) Synonyms concern the lexical level of the process; at this layer, the task is to retrieve domain terminology and synonymous terms. Outputs of these layers are applied as the basis in the concept formation. In the successive layers, we find tasks related to learning a concept hierarchy, relations, relation hierarchy and deriving axiom schemata. According to Cimiano, the higher levels represent "the most challenging task, as in principle there is no limit on the type and complexity of axioms and rules to be learned". Actually, when we apply a specific knowledge representation language - i.e., OWL – allowed axioms are constrained.

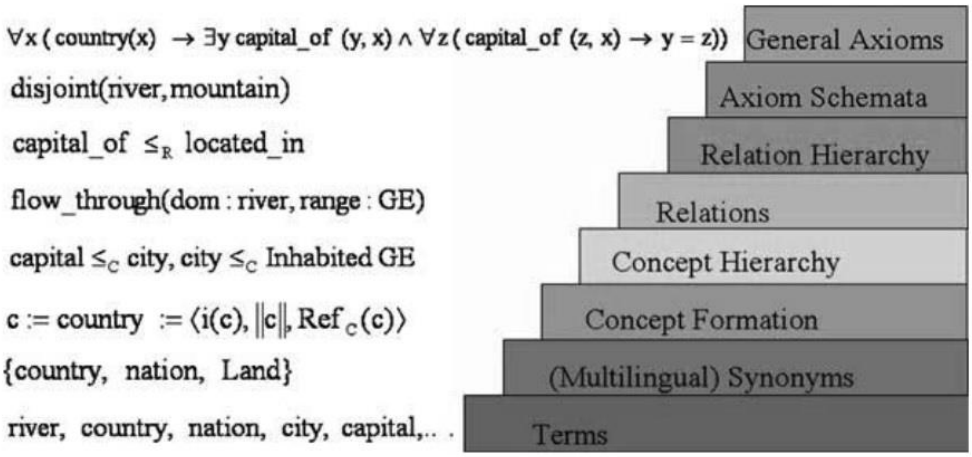


Figure 2.1 – Ontology Learning Layer Cake proposed by Cimiano (2006).

AS Lehmann & Völker (2014) also observe, Cimiano's layer cake has to be assumed as an ideal model, since it does not refer to a specific ontology representation language and it presumes a linear structure in a learning process. Another boundary seems to be the lack of distinction between syntactic and semantic features of ontologies, although the scheme is focused on lexical approaches, concerning OL from unstructured texts.

OL may be achieved through a manual development or a (semi-)automatic procedure. Manual ontology acquisition is basically used by knowledge engineering or domain experts, and it includes tools such as Protege-2000 (Grosso *et al.*, 1999; Noy *et al.*, 2000) and OntoEdit (Sure *et al.*, 2002). Besides, it represents a troublesome and time-consuming task, due to the fact that it

requires a precise knowledge of a specific domain and frequently results could be incomplete or inaccurate. Furthermore, “manually built ontologies are expensive, tedious, error-prone, biased towards their developer, inflexible and specific to the purpose that motivated their construction” (Hazman *et al.*, 2011).

In order to overcome issues coming from manual building procedures, researches are focused on the using of semi-automatic or full automatic ontology building methods. Such approaches try to achieve the task by applying two main ways: one based on the development of specific tools, the other on the employ of different information/knowledge sources.

OL may also be classified on the basis of the types of data from which conceptual knowledge is extracted, and subsequently systems learn, that is: structured, semi-structured and unstructured data.

Unstructured ones are the most difficult kind of data from which an ontology may be learnt. They require a processing phase more accurate than the one required by structured and semi-structured data. For this reason, systems able to learn an ontology from unstructured data frequently employ a natural language processor. Systems in this research area share NLP techniques, in the sense that they deal mainly with natural language texts. For such techniques, we may categorize three kind of NLP approaches, that is: automatic, semi-automatic and hybrid. For example, Sánchez Cisneros & Moreno (2004) propose a system which performs a shallow text processing with statistical analysis. Sabou *et al.* (2005) use a rule-based parser in order to identify dependency relations between words.

Due to the presence of several approaches for several applications in several disciplines, OL research domain seems hard to be outlined. In some cases, data sources, used to learn an ontology, differentiate systems; in other ones, algorithms and methods are applied to distinguish different lines of research.

Anyway, according to Lehmann and Völker (2014), such approaches may be classified in the following areas:

- **Ontology learning from text**. It deals with the extraction and population of an ontology using NLP and machine learning techniques.

An example is Read the Web[5], a research project at Carnegie Mellon University "that attempts to create a computer system that learns over time to read the Web". As well, since 2010, Never-Ending Language Learner (NELL), a never-ending machine learning system, aims at extracting facts from text in Web pages and at improving its reading competence. Therefore, the research goal is to create a KB extracting structured information from unstructured web pages. As stated by Carlson *et al*. (2010), "The thesis underlying this research is that the vast redundancy of information on the Web (e.g., many facts are stated multiple times in different ways) will enable a system with the right learning mechanisms to succeed". NELL uses a semi-supervised learning method, a set of KE methods and a KB representation to integrate outputs of such methods.

- **Linked Data Mining.** It intends to detect meaningful patterns inside RDF graphs, via statistical schema induction (Bühmann & Lehmann, 2012; Bühmann & Lehmann, 2013; Völker & Niepert, 2011) or statistical relational learning methods. This research area often applies clustering approaches in order to group interconnected resources.
- **Concept Learning in Description Logics and OWL** aims at "learning schema axioms, such as definitions of classes, from existing ontologies and instance data" (Lehmann & Völker, 2014). The basis for most methods in this area are Inductive Logic Programming and generic supervised machine learning approaches for description logics, e.g., DL-FOIL (Fanizzi *et al.*, 2008) and OCEL (Lehmann & Hitzler, 2010).
- **Crowdsourcing ontologies.** This approach, developed as an alternative to purely automatic methods, merges human intelligence, as well as the accuracy and capability of computer processing. In this area, one of the goals is inducing semantics of the tags used in social media data and folksonomies[6]. Using folksonomies to induce semantics, namely to

[5]http://rtw.ml.cmu.edu/rtw/.

[6]The term 'folksonomy' was introduced by Thomas Vander Wal (2005), as a blend word of folk and taxonomy. "Folksonomy is the result of personal free tagging of information and objects (anything with a URL) for one's own retrieval. The tagging is done in a social environment (usually shared and open to others). Folksonomy is created from the act of tagging by the person consuming the information. The value in this external tagging is derived from people using their own vocabulary and adding explicit meaning,

deal with an ontology, leads Van Damme *et al.* (2007) to introduce the term of 'Folksontology'[7]. Furthermore, people contribution is also used to perform, or take part in, tasks that computers are not able to accomplish efficiently[8]. An example of these systems is Amazon Mechanical Turk[9], in which humans are involved in HITs (Human

---

which may come from inferred understanding of the information/object. People are not so much categorizing, as providing a means to connect items (placing hooks) to provide their meaning in their own understanding.

In a few conversations around folksonomy and tagging in 2004 I stated, "folksonomy is tagging that works". This is still a strong belief the three tenets of a folksonomy: 1) tag; 2) object being tagged; and 3) identity, are core to disambiguation of tag terms and provide for a rich understanding of the object being tagged".

Source http://vanderwal.net/folksonomy.html.

[7]"This branch of ontology deals with the intersection between highly structured taxonomies or hierarchies and loosely structured folksonomy, asking what best features can be taken by both for a system of classification. The strength of flat-tagging schemes is their ability to relate one item to others like it".

Source: https://en.wikipedia.org/wiki/Folksonomy.

[8]See Chilton *et al.* (2013) and Karampinas and Triantafillou (2012).

[9]http://www.mturk.com/.

Sample Source: https://www.mturk.com/mturk/welcome?variant=worker.

Typically, Amazon Mechanical Turk cases of uses are:

- Processing photo/video
- Data cleaning/verification
- Information collection
- Data Processing.

In Amazon Mechanical Turk users are distinguished in Workers o Requesters. Workers are employees who work on HITs; Requesters are employers (companies or independent developers) who ask for the accomplishment of a specific task. Requesters will be using the results of this task within their processes and systems.

We give hereby some examples of HITs that Amazon provides to workers:

- Select the correct spelling for these search terms
- Is this website suitable for a general audience?
- Find the item number for the product in this image
- Rate the search results for these keywords
- Are these two products the same?
- Choose the appropriate category for products
- Categorize the tone of this article

Intelligence Tasks), such as writing product descriptions, answering other users' questions, choosing the best among several photographs of a storefront, or identifying performers on music CDs.

In Table 1, we schematize a list of OL approaches, which vary as for the source data used, as well as for the data sources employed:

Table 2.1 - Schema of different OL approaches.

Data Types	Data sources	Methods
Unstructured	Text	Statistical/Stochastic Approach
		Pure NLP Approaches
		Integrated Approaches
Semi-structured	Linked Data	Data Mining Approaches
	Description Logics and OWL	Web Content Mining Approaches
Structured	Crowdsourcing	Hybrid Approach

Several researches present statistical approaches to OL tasks and different methodologies have been developed for each task. In our dissertation, we deal with the extraction and population of an ontology from unstructured texts using NLP and machine learning techniques.

3. Ontology Learning from Texts

The first distinction among different text understanding techniques is represented by the use of shallow or deep semantic analysis. By the way, it is worth to remember that some recent approaches apply hybrid methods, in order to exploit all the benefits derived from both methods and for each step of their processes.

The shared goal is retrieving important concepts and relationships among them from the extracted knowledge. Due to this aim, attempts have

---

- Translate a paragraph from English to French.

On the other hand, Requesters may categorize and classify items (taxonomy construction), train algorithms, and also collect and understand sentiment on their data (through the Sentiment App).

Source: https://en.wikipedia.org/wiki/Amazon_Mechanical_Turk.

been focused on the development of adequate tools mainly employed in OL tasks.

Generally speaking, "the shallow semantic analysis measures only word overlap between text and hypothesis" (Bos & Markert, 2005). This means that starting from a hypothesis about words, which stand for concepts and their relationships, the shallow semantic analysis computes the distance between such a hypothesis and words in a given text. In fact, starting from a tokenization and lemmatization of text and hypothesis, this analysis uses Web documents as a corpus and assigns inverse document frequency as a weight to each entry in the hypothesis. Thus, we have a higher score for those words that occur less in the text; this means that we assign more importance to less frequent words. Shallow analysis needs tagged corpora as training resources. This technique may be applied at both the syntactic and the semantic level. Regarding syntactic analysis, and in order to generate partial analyses of sentences, shallow NLP applies methods and tools, such as chunkers – dividing sentences into chunks (Abney, 1992) – and parts-of-speech taggers – assigning a syntactic label such as NP to a chunk).
Shallow approach is largely used in various tasks of OL, as for instance:

- Term Extraction: is the first goal of shallow NLP techniques. Usually, terms are extracted using chunkers[10]. Outputs, as NPs, may be include in the basic vocabulary of the domain. Usually, in order to evaluate weight of extracted terms with respect to the corpus, statistical measures of IE are applied, such as TF*IDF algorithm (Salton & Buckley, 1988).
- Taxonomy Extraction: this task is related to the extraction of hierarchical relations among ontology classes or individuals (Staab & Mädche, 2001; Cimiano & Völker, 2005). The hierarchy is usually extracted using lexical and syntactic patterns, expressed by means of regular expressions.
- Relation Extraction: using shallow parsing, it is possible to extract only limited reliable relations, which means simple patterns such as

[10]A chunker, or shallow parser, analyses a sentence and identifies its constituents divides plain text into sequences of semantically related words, providing the syntactic structure of a sentence. Opposed to full parsing, chunking is based on a limited tree depth.

> NP+VP+NP. Indeed, this analysis does not cope with complex sentence structures in which there are long-distance relationships, disjunct and long-distance dependencies, or other language ambiguities. By the way, this limit does not allow axiom learning, obtainable only with deeper syntactic methods.

While shallow NLP covers syntactic steps in the learning process, other different methods are also applied to generate a shallow semantic parsing (or semantic tagging). These methods are more useful in the ontology population procedure than in the learning one (Etzioni *et al.*, 2004; McDowell & Cafarella, 2008), because they rely on conceptual structures in order to guide extraction. Being devoted to extract entities and their relationships**,** shallow semantic parsing approach is extremely different from the one based only on texts and syntactic NLP. Indeed, shallow semantic parsing requires the identification of structures which describe the context of entities, together with the relations among these. Consequently, the population process, achieved by means of shallow semantic parsing, depends on a set of knowledge resources, such as frame, templates or roles. Such these knowledge resources also hold semantic information, which define meaning contexts and may be suitable for discovering instances and relations. According to (Giuglea & Moschitti, 2006), these resources may include role taxonomies, lists of named entities and also lexicons and dictionaries. The reason such resources are key elements is that shallow semantic parsing aims at achieving a word sense disambiguation process. This disambiguation process allows to match a given word to the correct meaning or concept it sends back to within specific sentence contexts or propositions.

This procedure may be also applied to recognize particular semantic relationships, such as synonyms, meronyms, or antonyms, using predefined patterns to which terms should conform to. In other words, these patterns are applied to control productive functions which concern entire classes of words or single, isolated words.

While shallow semantics may adequately response to some ontology learning steps, it results inadequate for tasks that are more complex. This is because shallow methods do not guarantee a fine-grained linguistic analysis. Besides, dealing for instance with anaphora resolution, quantifier scope resolution, and so on, requires a text processing in order to extract rich domain

ontologies. Due to the fact that deep NLP allows to work not only on concepts but also on relations and axioms, such approach seems more appropriate for understanding the meaning of sentences and discourses. Indeed, if shallow methods focus only on text portion, deep ones allow to obtain a fine-grained analysis, working on the whole meaning of a sentence or a discourse.

Deep methods represent a useful approach to extract representations and to infer on the basis of such representations. It means that this kind of analysis may contribute to inferencing and reasoning capabilities of machines through textual Web resources representation based on a machine-readable standard ontological language.

Due to the need of applying an ontological language, i.e. a formalized language, in order to process textual resources it is necessary to use grammar rules.

Such set of grammar rules may be applied by a syntactic parser, which is "the first essential component for a deep analysis of texts" (Zouaq, 2001:5). Indeed, syntactic parsing uses a set of grammar rules, known as syntactic grammars, in order to assign parse trees to sentences.

Noticeably, as we stated in Chapter I, a formal language and its syntactic grammar rely also on a vocabulary, which means on all the acceptable combinations of characters which compose the specific alphabet. Such predefined vocabulary may be used in parsing sentences[11].

Another way to create the lexical KB useful to parse a sentence rely on the adoption of training sets of hand-labelled sentences. This methodology represents the foundation of statistical parsers (Klein & Manning, 2003; Jurafsky & Martin, 2009). Statistical parsers, or dependency parsers, are developed on the assumption of Dependency Grammar (DG), which states that linguistic units may be connected to each other by direct links. As we will see in Chapter IV, the foundations of DG may be traced primarily in Lucien Tesnière's works[12].

Parsing produces outputs exemplified in the form of phrase structure trees representations or dependency parses. Indeed, "phrase structure parses

[11]In the chapters which follow we present some of these linguistic resources.

[12]Tesnière developed the concept of valency in detail, and the primary distinction between arguments (actants) and adjuncts (*circumstants*, French *circonstants*). We will discuss Tesnière's theory in Chap. IV.

associates a syntactic parse in the form of a tree to a sentence, while dependency parses creates grammatical links between each pair of words in the sentence" (Zouaq, 2011:5).

Most of syntactic theories apply both formalizing methods to such approaches, which are to be seen as complementary and not in opposition. Also in shared tasks, as those exemplified in CoNLL (Conferences on Natural Language Learning) meetings, many scholars (Briscoe *et al.,* 2006; Zouaq, 2008; Kübler *et al.*, 2009), apply dependency parsing, because it allows to model predicate-argument structures in a more intuitive way.

Indeed, using predicate-argument structures for extraction paradigms seems to allow the reaching of high quality IE results (Surdeanu *et al.*, 2003).
Various researches are aimed at establishing a correspondence between predicate-argument structure and first order predicate logic, even if this goal has been found to be problematic (Hurford *et al.*, 2003). Also according to Luuk (2009), "the predicate/argument system of natural language is more complex than that of first order predicate logic". Thus, a complexity is caused by the fact that the predicate-argument structure allows to use the same kind of term in order to fill both the argument and the predicate slot.
For instance, the term *cat* may occur into sentence as follows:

(1) The **cat** meows (argument)
(2) Sid is a **cat** (predicate).

On the contrary, in first order predicate logic the same term cannot fill both predicates and arguments.

Anyway, Surdenau *et al.* (2003) state that those systems, which label predicate-argument structures on the output of full parsers, are central for this way to perform IE from texts. Indeed, the authors argue that if "we know (a) predicates relevant to a domain; and (b) which of their arguments fill *templette*[13] slots" (*Ibidem*), we may develop a domain-independent IE paradigm. We will examine in depth template-based IE in Paragraphs 7 and 8.

[13] *Templettes* drive the identification and selective extraction of relevant information. Indeed, in event templettes, event basic information (e.g., main event participants, event outcome, time and location) are represented with slots in frame-like structures (Surdenau *et al.,* 2003).

Most of ontology learning methodologies apply syntactic parsing, based on patterns or machine learning, to improve the extraction of relevant structures. It means that syntactic parsing may allow a fine-grained analysis, guaranteeing also the extraction of both Atomic Linguistic Units (ALUs) and relations plus axioms learning. We will present works based on this approach in the following chapter.

Applied method must be adequate to the particular task we want to perform: for instance extracting a whole ontology, or only a constituent of such ontology, (classes, relations or axioms).

The most common tasks, which characterize IE systems and are based on the classification provided by the seventh MUC (1998) are represented by[14]:

1. Named Entity recognition and classification (NER/NERC) - finds the entities in the text
2. Reference Resolution - finds identities between entities
3. Template Element construction (TE) - finds entities attributes
4. Template Relation construction (TR) - finds the relations between entities
5. Scenario Template construction (ST) - finds the events in which entities participate.

For each task, the quality of the results depends on the quality of the results obtained by the task performed immediately before; in this way, a so called 'snowball effect' is produced. Thus, NERC quality of the results is usually very high, while the Scenario Template task produces a lower one. In the following, we give a more detailed description of the tasks, mostly based on Chinchor & Robinson (1997) and Cunningham (1999)[15].

[14]We use a classification that is a little bit different from the one provided by MUC. Indeed, MUC describes NERC as Named Entity Recognition (NER) and Reference Resolution as Corefence Resolution (CO).

[15]In Muc-6, Grishamn and Sundheim (1996) take account of three MUC tasks in order to measure aspects of an IE or language understanding system. Such tasks, collectively called SemEval (Semantic Evaluation) (MUC-6), are:

- Coreference: coreferential NP have to be marked by the system.

IE tasks may be also achieved using an ontology to aid the IE process, while the IE results may be used to contribute to populate the ontology. Therefore, ontologies may be applied to help the extraction of terms and relations from semi or unstructured documents.

In the next chapter, we will introduce the most used algorithms which have been developed for such tasks. Actually, various methodologies have been applied to increase retrieval and extraction system performance in different knowledge domains. The common aim is to process unstructured texts and, through semantic annotation procedures, formalize them in a structured representation. This step of converting texts represents the way in which we move to machine-readable language with the purpose to systemize, manage and extract knowledge from a given amount of data. Subtasks involved in the formalizing process, concern entities and relations between them and their attributes. It means that in a text we have to analyse not only subjects and objects, which take part into a specific situation, that is to say discourse and sentence contexts, but also identify which kind of relation exists among them. Reconstructing the network of relations and attributes among entities leads us to reconstruct the Aristotelian definition process concerning concepts. Thus, we get close to understand the meaning expressed in a text. As a matter of fact, as we will demonstrate in the following chapters, such meaning may be analysed through a precise formalization of natural language, which means a formalization methods based on linguistic studies rather than on the development of stochastic algorithms.

4. Term Extraction and Named Entity Recognition and Classification (NERC)

In order to recognize and extract main entities form a text, the first task to consider is Named Entity Recognition and Classification (NERC). The term Named Entity was coined for the Sixth Message Understanding Conference (MUC-6) (Grishman & Sundheim, 1996).

- Word sense disambiguation: the system has to determine the sense of each open class word (noun, verb, adjective, adverb) using Wordnet synsets
- Predicate-argument structure: creation of a tree interrelating the constituents of a sentence, applying some set of grammatical functional relations.

Since the early '90s, the research field related to NERC has been characterized by a growing interest and by the development of different methodologies, going from handcrafted rules to machine learning approaches. We refer to the definition of NERC given by Bengfort, which is:

> *More formally, the task of Named Entity Recognition and Classification can be described as the identification of named entities in computer readable text via annotation with categorization tags for information extraction* (Bengfort, 2013).

Thus, this task aims at finding entities (e.g. nouns, proper names, noun phrase) and classifying them in pre-defined categories as person names, places, organizations, etc. Samples of named entities are represented by proper names, surnames, geographic locations, ages, addresses, phone numbers, names of companies, of streets and addresses.

According to DARPA's MUC, NERC top-level categories are traditionally identified in three types:

1. Entity Names, in which we insert Organizations, Persons and Locations
2. Temporal Expressions, that includes Dates and Time
3. Number Expressions, which refers to Money and Percent (Chinchor & Robinson, 1997).

Noticeably, many subcategories may be defined additionally to these basic ones, such as Distance, Speed, Age, Weight, City, State, etc., depending on the specific field from which we have to extract entities. This means that knowledge representation, we are interested in, drives the definition of subcategories on the basis of the domain entities we need to extract and classify.

Entity Names category is used to describe people, locations, geopolitical bodies, events, and organizations; thus, it represents the main category on which most of researches are focused on. The reason of such an interest is motived by the need of extracting subjective information related to these categories. Therefore, such an IE is oriented towards the so-called sentiment

analysis, which means opinion mining, in order to analyse leanings concerning politics, products and so on.

Named entities are often formed by compound words, e.g. University of Salerno or Federal Reserve System, and by acronyms (UNISA, USA, LADL); thus, they require other tools to be recognized. In order to deal with such named entities, usually an external KB and name lexicons are applied by many approaches and methods. In other words, it is necessary to develop gazetteers[16], or other similar resources, which hold information allowing to recognize all the tokens related to a given entity.

For these reasons, NERC also entails the use of prediction models, during parsing or chunking process, in order "to predict whether a group of tokens belong in the same entity" (Bengfort, 2013).

Usually, NERC is based on a part-of-speech tagger and a noun phrase chunker, beyond a capitalization classifier.

NERC requires an inference process to clearly recognize entities that refer to the same name, or other items that have similar attributes. The inference process is useful for determining if a chunk is a named entity or for classifying a named entity in the correct category, especially when we have ambiguity. For example, this is the case of the entity *Washington*, and similar ones, that could refer to a person or to different type of locations. In order to

[16]A gazetteer is an alphabetical list of place names with information that can be used to locate the areas that the names are associated with. There are three styles of gazetteer: alphabetical list, dictionary, and encyclopedic.

The alphabetical list includes place names and locations, information typically found in atlases, and a minimal indication—often latitude and longitude coordinates or some other Cartesian-coordinate scheme—of where it is to be found on any accompanying map(s).

Dictionary-style gazetteers include location information in the form of geographic coordinates or descriptions of spatial relationships to other places. Entries may also include a pronunciation guide and limited information on demographics and history. Because gazetteers are not able to cover an area in its entirety, most gazetteer preparers will include a preface describing how inclusion decisions were made.

Encyclopedic gazetteers will include all the information of a dictionary-style gazetteer but the information will be more detailed and may come in the form of articles written by area specialists. Information will be expanded but the scope will still be limited. These gazetteers should also include a statement explaining those limitations.

(Source: University of Illinois at Urbana-Champaign.
http://www.library.illinois.edu/max/collections/gazetteers.html)

solve such ambiguity, usually machine-learning systems have been developed, performing analysis on the basis of Maximum Entropy Models, Hidden Markov Models and other statistical methods[17] (for examples in Florian *et al.*, 2003; Chieu & Ng, 2002; McCallum & Li, 2003)

In recent years, NERC has been dedicated to extract and classify entities in Social Media websites, such as Facebook and Twitter. Particularly, traditional NLP tools deal with a lower performance in NERC, when processing tweet corpora due to the fact that such tools generally are trained on news corpora. Indeed, especially Twitter, due to the limit of 140 characters, lays down "a new and challenging style of text for language technology due to their noisy and informal nature" (Ritter *et al.*, 2011). The main issue is the lack of background knowledge, which means that tweets do not present a context sufficient to determine the entity type. Nevertheless, NERC also represents a key task for reference resolution, disambiguation and meaning representation.

5. Reference Resolution

In this paragraph, we introduce the issue of reference resolution in text extraction, which is the identification of identity relations between Named Entities (Jurafsky & Martin, 2009). Usually, scholars analyse two kinds of reference in texts: coreference and anaphora. In our opinion, the problem may be tackled as a whole, even if it presents various linguistic features in both the expression ways and the sentence structures it produces.

Coreference resolution (CO), also called pronominal coreference, has the goal of finding which entities and references (e.g., pronouns) are identical, which happens when such elements refer to the same linguistic item(s).

(1) **John** drove to **Judy**'s house. **He** made **her** dinner**.**

On the contrary, Anaphora Resolution (AR) deals with the problem of resolving references to earlier or later items in the discourse.

(2) **Max** helped **Mary**. **He** was kind and **she** was grateful.

[17]For more information see Chapter III.

(3) When **he** arrived at home, **John** went to sleep.

In Linguistics, when we refer to a previously introduced entity, we cope with an anaphora (2). While, when the pronoun refers to an entity which occurs later, we deal with a cataphora (3).

Anyway, in our opinion the distinction between anaphora and cataphora is not noteworthy inasmuch same techniques are required in order to solve these phenomena. Indeed, according to our theoretical and methodological approach, which will be introduced in the next chapters, these phenomena may be solved just by means of an accurate lexical and semantic analysis.

In the samples provided, anaphora use has a lower impact than in other languages. Actually, the presence of the pronouns *he* and *she* helps us to distinguish that *Max was kind*, while *Mary was grateful*. On the contrary, some languages, such as Italian, are typical prodrop ones, which means that these languages may drop pronouns presenting only the VP, like in (2a).

(2a) Max uscirà presto. Chiamalo prima che Ø sia irrintracciabile**.**
**** Max** will go out in a little while. Call him, before (**he**) is unavailable*.

This feature entails that, during text recognizing, we may have not the referring expression.

Dropping pronouns is defined as a diexis. Generally, the most common categories involved in such phenomenon are those of person, place and time.
More contextual information are required in order to identify the meaning and the function of the elements involved in a diexis. In spoken language, we may refer to extra-linguistic situation for overcoming the information gap. In written context, the ambiguity is higher, especially in prodrop languages, which means that we may only analyse sentence contexts to resolve a diexis.

For these reasons, recognizing and resolving AR is defined as a very challenging task in NLP (Mitkov, 1999; Denber, 1998).

In IE task, the most spread problem of anaphora is Nominal Anaphora, which is the use of pronouns to refer to a noun or an NP.

Even if Nominal Anaphora, such as (2) and (3), is the most investigated form of anaphora, in our opinion, also verbal and adverbial anaphoras present an ambiguity hard to solve.

(4) Please, **fill out the form**. Failure to **do so** will result in this application being rejected. (verbal anaphora).
(5) She put the bag **on the table**. It should still be **there**. (adverbial anaphora).

In these phenomena, the relation between the anaphor and the referent (antecedent or subsequent) may be implicitly or explicitly expressed. In the first case, implicit expression, the anaphor and the referent do not stand in a structural or grammatical relationship. On the contrary, in explicit expressions, a relationship between the anaphor and the referent is present and may be used to resolve such an ambiguity.

6. Relation Extraction

Relation Extraction represents a further step in IE, and is typically applied after NERC and reference resolution, which is also useful to analyse a text and to turn unstructured information into structured ones. This task aims at gathering relations between named entities, establishing which are meaningful for the concrete application.
Culotta *et al.* (2006) define relation extraction as:

> *(…) the task of discovering semantic connections between entities. In text, this usually amounts to examining pairs of entities in a document and determining (from local language cues) whether a relation exists between them.*

Different techniques are applied in order to recognize and extract such relations, e.g. knowledge-based, supervised and self-supervised approaches. Each methodology does not deal with each kind of relations that may occur between entities. For example, hyponymy, that is the relation between *author* and *Shakespeare*, or *England* and *European country*, is easily recognized by knowledge-based methods. However, such methodology does not work adequately with the extraction of other kind of relations, like meronymy, which denotes a constituent part of or member of something, e.g., *finger* is meronym of *hand.*

Usually, knowledge-based approaches are applied in domain-specific tasks, in which texts are similar and a closed set of relations needs to be identified (Konstatinova, 2014).

According to Riloff & Jones (1999) and Pasca (2004), knowledge-based systems rely on pattern-matching rules manually crafted for each domain. Noticeably, not all relations may be considered domain-dependent, and some of these refer to an open domain. In such cases, lexico-syntactic patterns used to extract relations are not suitable to uniquely identify the given relation (Hearst, 1992).

7. Template Element Construction (TE)

In IE, templates are used as extraction patterns in order to retrieve relevant information from documents (Muslea, 1999).

In template-based IE, standard algorithms "require predefined template schemas, and often labelled data, to learn to extract their slot fillers (e.g., an embassy is the Target of a Bombing template)" (Chambers & Jurafsky, 2011). Templates are used for more articulated representation of a knowledge domain than atomic facts which are the object of learning on which relation discovery is focused (Banko *et al.*, 2008; Carlson *et al.*, 2010).

The use of templates is suitable to learn information from texts and may "alleviate the main bottleneck in creating knowledge-based systems that is 'the extraction of knowledge'" (Vargas-Vera *et al.*, 2001). Basically a template is a simplified knowledge structure, provided with little inferencing capabilities; this IE model has been developed since the early '90s (Appelt *et al.*, 1993).

Template Element construction (TE) task aims at identifying additional information about template entities, e.g. their attributes.

Among various kinds of parsing approaches in IE systems, which also include partial parsing and grammatical parsing with common-sense understanding[18], the template-based one may be considered an intermediate type.

[18]IE systems, which apply partial parsing, recognize syntactic structures without generating a complete parse tree for each sentence. Advantages of such systems may be distinguished in greater speed and robustness. These characteristics allow to process a large set of documents and to deal with unstructured and informal texts.

The template-based approach, proposed by Riloff in 1996, recognizes texts using extraction patterns and semantic features, associating them to slots in a predefined frame. Generally speaking, in templates, the main verb represents the trigger in any tense, which can be reliably identified using linguistic rules.

Riloff provides an example of frame for the event 'murder' in the terrorism corpus of MUC as target domain (Latino-American terrorist domain):

```
Name-frame:
      Event-type murder (active verb murdered)
Slots
      Victim:
            <subject> (human)
      Perpetrator:
            <prepositional-phrase, by> (human)
      Instrument:
            <prepositional-phrase, with> (weapon)
```

Thus, while the Name-frame identifies the event type which is associated with a verb, i.e. murdered, slots are used to define co-occurring elements in sentence contexts. Such elements describe all entities involved in each event, for instance *Victim*, *Perpetrator, Instrument,* combining each slot with a class such as *human* or *weapon.* Slots and related classes are defined on the basis of patterns which outline linguistic and semantic rules.

Therefore, starting from semantic and lexical features of a given verb and its co-occurring elements, templates are developed by means of which MUC corpus is annotated.

8. Template Scenario Production (ST)

On the other hand, the second kind of parsing, the Cyc approach (Lenat et al., 1985), attempts to integrate a knowledge base of common sense with different components, such as inference engine, representation language, etc. Cyc Knowledge Server aims at providing a deep layer of understanding; such layer can be useful to make other programs more flexible (Vargas-Vera *et al.,* 2001). For more information, see www.cyc.com.

Template Scenario (ST) production task performs the extraction of events with various entities playing a role and/or being in certain relationships.

An event is an occurrence, described as a change of state, that happens and in which participants are involved.

Therefore, the aim of such task is filling in the event template with specified entities and relationships.

"Because manually creating templates can be tedious and time consuming, several researches have worked on automatically extracting templates from training examples that have been pre-processed" (Ong, *et al*., 2008).

Representation of an event may be described using different approaches; in this dissertation, we present two kinds of these:

- Template in MUC;
- Verb in ACE.

The differences between MUC and ACE approaches are represented by the level of source granularity, e.g., they are sentence-based, single-document or multi-document, and it is assumed that they may be applied to documents, e.g., to a document that contains a single event or more than a single event.

Thus, MUC approach may be defined as a template-based event IE, in which one document represents one event. On the other hand, ACE approach calculates that a verb represents one event, each event type being as a set of possible arguments roles, which may be filled by entities, value or times, and that one document has many events.

In the template, four kinds of slots exist:

1. Set fill: by selection from a pre-specified list of categories defined in the fill rules for a given slot.
2. String fill: with an exact copy of a text string from the input.
3. Normalized fill: with a text string that is converted to a canonical form in accordance with the filled rules for a given slot.
4. Index fill (pointer): with the index of an object <> (Li F., 2015).

The main issues for template based IE is related to being domain independent. This means that organizations and persons with some associated

attributes may be extracted from texts without knowing what roles or relationships those elements may enter into. Therefore, being domain independent is a characteristic which really limits the KE process, due to the fact that domain information may be used to constraint extraction[19].

9. System Classification

In recent times, many different systems and frameworks have been proposed for OL tasks. Most spread and mentioned tools are OntoLearn (Velardi *et al.,* 2005), OntoLT (Buitelaar *et al.*, 2004), Terminae (Aussenac-Gilles *et al.*, 2008) as well as TextToOnto (Mädche & Volz, 2001) and its successor Text2Onto (Cimiano & Völker, 2005).

Usually, performance evaluation of these five tasks – NERC, Reference Resolution, Relation Extraction, TE and ST – represent the basis for IE system comparison. Some tasks are easier to accomplish, others require an improvement of current methodologies. In the last years, for more spread languages, such as English, extraction performance has improved, mainly in NER, reaching an accuracy of more than 95%. Template element construction and template scenario production are characterized by a lower level of accuracy, respectively of about 80% and 60% (Cunningham, 2005).

Noticeably, the difference in system performance also depends on the texts processed, e.g. domain texts with a specific telegraphic style, as the medical one, are simpler to analyse, mainly during reference resolution tasks.

In order to describe IE systems, beyond their performance in precision and recall, we have to analyse the structure on which they are based. Indeed, each system is characterized by its own sets of functions and modules.

According to Hobbs (1993), "an information extraction system is a cascade of transducers or modules that at each step add structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically".

Starting from this statement, Hobbs defines an IE system analysing:

- Transducers (or parsers) that compose it;
- System input and output;

[19]For more information, see Chapter V.

- Kind of structure it added;
- Type of lost information;
- The form of rules applied;
- The way in which rules are applied;
- The way in which (new) rules are acquired.

Traditionally, on the basis of such features, we may distinguish some functions and modules in which a system is structured.

Regard transducers that compose systems, in Chapter I we already gave an overview of the different kinds used in formal and natural languages processing.

Parsers may be based on different frameworks, according to the methodology used for reconstructing the knowledge context from which we have to extract entities, concepts and axioms. Usually, such module takes a sequence of lexical items or a phrase as input, in order to produce a parse tree for the entire sentence, as for example the one developed by the University of Stanford, which is a high-performance neural network dependency parser[20].

"Most recent academic research in this area starts from the assumption that statistical machine learning is the best approach to solving information extraction problems" (Chiticariu *et al.*, 2013).

Chiticariu *et al.* analyse published research papers from 2003 to 2012, taken from EMNLP, ACL, and NAACL conference proceedings, focusing mainly on the topic of entity extraction, as most industrial systems offer this feature. Classifying these papers, they identify three kinds of techniques used in this task: purely rule-based, purely machine learning-based and hybrid of the two. As shown in their analysis, generally, academic papers, also when they develop rule-based systems, are inclined to obfuscate the use of rules, "emphasizing the machine learning aspect of the work" (*Ibidem*). Some examples are cited, in which authors do not acknowledge rule application, preferring periphrases such as "dependency restrictions" (Schmitz *et al.*, 2012), "entity type constraints" (Yao *et al.*, 2011), or "seed dictionaries" (Putthividhya & Hu, 2011).

They also propose a survey on commercial entity extraction systems from fifty-four different vendors listed in (Yuen & Koehler-Kruener, 2012). In the industrial world, the survey has been conducted on analyst reports and

[20]For more information, see http://nlp.stanford.edu/software/nndep.shtml.

product literature produced until 2013. Results show that in the industrial world the trend of techniques applied is opposite to that of the academic world, not even reflecting it. Indeed, commercial products are mainly rule-based, "with large vendors such as IBM, SAP, and Microsoft being completely rule-based" (Chiticariu *et al.*, 2013). A disconnection between research and industry efforts seems evident, even if the two communities interact with each other.

Chiticariu *et al.* suppose that the reason of such difference between these two worlds may be based on different evaluations of IE system benefits and costs. Authors observe that the risk is the complete separation of industrial and academic products. Thus, while commercial systems response to specific customer needs with ad-hoc solutions, "researchers pursue ever more complex and impractical statistical approaches that become increasingly irrelevant".

Otherwise, as stated by Mendel (2013), due to the growing amount of data available on the Web and the need of Big Data analytics, IE is becoming a big business. For these reasons, IE tasks mainly rule-based ones are still an open research field, as well justified in the industrial world.

This gap seems unbridgeable, if researchers do not start to (re)consider the opportunity of developing more efficient and effective rule-based systems. Beyond these considerations, in our dissertation we will present both statistical and rule-based methodologies, underlying their boundaries and opportunities. In this way, we will lay the foundation to present our theoretical and practical framework on which we base our proposal, in order to achieve ontology-learning tasks.

In this chapter, we have been dealing with the analysis of specific tasks in OL, and of different methodological frameworks developed to improve results in such field. As we mentioned, OL is a complex IR and IE assignment and a correct KR seems indispensable in the achievement of both ontology acquisition and population. Semantic representation, accomplished using specific standardized languages as well as shared conceptual models, and natural language formalization, based on an accurate linguistic observation, become the keystone in different methods. Each of these methods is characterized by "the semantics of its input data and the type of inference that can be performed to derive new ontological knowledge from these data". (Völker, 2009).

Shamsfard & Barforoush (2004) distinguish ontology-learning techniques in statistical, linguistic and logical approaches. Indeed, various OL applications have been, and are, build up, being a "research field at the intersection of machine learning, data and text mining, natural language processing and knowledge representation" (Cimiano *et al*., 2009).

In chapters which follow, we will present some of the main applied techniques and tools, both stochastic and rule-based.

III - ONTOLOGY LEARNING AND POPULATION BY STOCHASTIC METHODS

In this chapter, we deal with ontology learning and population achieved by means of stochastic methods, yet without using traditional subdivisions in this specific field. Such a choice is justified by the presence of many hybrid approaches, regarding both data sources, i.e. KBs, applied methodologies and ideal views of the information space[1]. Anyway, for each of the technique presented, we will provide an accurate description, together with some samples of specific applications.

In his analysis, as for OL process, Omelayenko (2001) individuates some tasks in which ML techniques may be applied:

- **Ontology creation**. Machine Learning (ML)[2] methods attempt to discover the most important connections within the domain analysed; they also may check and verify constructed knowledge basis. In addition, Ontology acquisition by means of IE has been investigated since latest '90s. Generally speaking, such methods are developed using a semi-automatic approach based on machine-learning techniques (Mädche and Staab, 2000; Kietz et al., 2000; Craven et al., 1999).
- **Ontology schema extraction.** In such task, ML is performed to process data and meta-knowledge (i.e., meta-ontology) in order to provide a ready-to-use ontology as an output.
- **Extraction of ontology instances**. It is very similar to IE task and page annotation. Therefore, ML techniques are applied to populate a given ontology schema and to extract instances from data.

[1]Information space is the set of concepts and relations among them held by an information system; it describes the range of possible values or meanings an entity can have under the given rules and circumstances.

Source: https://en.wikipedia.org/wiki/Information_space.

[2]The classical definition of ML, by Mitchell (1997), states what follows:

> *The system learns from some experience E according to some performance measure P if it improves its performance (as measured by P) after passing the experience E.*

- **Ontology integration and navigation**
- **Ontology update**
- **Ontology enrichment**

In the following, we propose an overview about the stochastic methods used in NLP which are particularly dedicated to these tasks[3]. Most of such methods have been developed for scientific domains, such as Biology, Medicine, and so on. However, we believe that these may be considered as sample approaches, even if our application field is different. Indeed, beyond the (meta)data formalisms and conceptual models used, these different methods can be easily applied to any knowledge domain.

1. Distributional Semantics

In this paragraph, we introduce the concept of distributional semantics, and we defer to the next ones for the analyses of the most used techniques in this specific field. Usually, these approaches apply distributional similarity[4] models, developed on the assumption that two words are semantically similar when they have the same linguistic distribution, i.e. when they may appear in the same context(s) of word(s), occupying the same position(s). Thus, the study of meaning is dealt with corpus-based computational methods. This means that traditional stochastic approaches to distributional semantics observe the context in which words appear in order to deduce structures and organizations of semantic representations. Therefore, observed syntagmatic distribution in texts represents combinatorial behaviours to which lexical properties have to be reduced. Two linguistic expressions have to be assumed as similar when the contexts in which they can occur are similar.

Base hypotheses, coming from various approaches and included in the so-called distributional semantics, assume that semantic representation can be inferred from word co-occurrences in large text corpora. Such co-occurrences are measured through the statistical distribution of words inside

[3]In KE tasks we include NERC, Ontology Learning and Population, anaphora resolution and all other specific tasks that we have analysed in the previous chapter.

[4]Semantic similarity concerns different types of semantic relationships, characterised by different logical properties and inferences.

sentence/discourse contexts. Lenci (2008) defines this prospective on meanings as a 'usage-based' one, in which word semantic behaviours are characterized by text distribution. This group of approaches is fairly rich, which justifies the use of different terms to define it, that is distributional, context-theoretic, corpus-based or statistical.

The Distributional Hypothesis (DH), stated as above, may be clearly connected to post-bloomfieldian American structuralism, especially the one by Zelig S. Harris. Indeed, Harris (1951) maintained that the typology of the whole language might be obtained and explained through the distributional study of the behaviour of its entities. As we have seen in the previous pages, entities of constituents within a language are linguistic units belonging to the dictionary (alphabet) of such a language. So, if we find two linguistic units, w_1 and w_2, which have similar distributional properties, e.g., which occur together with the same unity w_3, then we can include them inside the same linguistic class.

Harris first introduced his proposal for phonemic analysis; then, he applied the concept to other linguistic levels:

> *In both the phonologic and morphologic analyses the linguist first faces the problem of setting up relevant elements. To be relevant these elements must be set up on a distributional basis: x and y are included in the same element A if the distribution of x relative to the other elements B, C, etc. is in some sense the same as the distribution of y. Since this assumes that the other elements B, C, etc., are recognized at the time* (Harris, 1951:7).

Therefore, according to Harris, the "linguistic environment" represents the basis on which to measure the degree of similarity of two words. Indeed, in most sentence contexts, the interchangeability between w_1 and w_2 demonstrates the existing similarity between these two words (Harris 1954:157).

Thus, according to Harris, syntagmatic relationships among words represent the basis on which to analyse lexical paradigmatic relations.

Through the same process, Harris explains that word meaning identity depends on the occurrence in some sentence contexts:

> *It may be presumed that any two morphemes A and B having different meanings, also differ somewhere in distribution: there are some environments in which one occurs and the other does not* (Harris, 1951:7 note 4; see also Harris 1954).

In this way, Harris discovers also the possibility of performing semantic analysis using word distribution. He derives the concept of meanings as an explanans of linguistic theory from Bloomfield. Nevertheless, while Bloomfield considers semantics as not analysable through linguistic study, because "the linguist cannot define meanings" (Bloomfield, 1933:145), Harris affirms that the distributional approach may be also applied to explanans of meaning.

This statement may be at odds with the presence of NL idioms and complex expressions[5], in which meanings cannot referred to as an amount of simple words. It means that such linguistic expressions have to be processed as a *unicum* characterized by a semantic charge which differs from the semantic charge of single simple words. Even if Harris' substitution principle seems not suitable for analysing these expressions, compositionality is still an important issue in the agenda of distributional semantics nowadays (cf. also Jones & Mewhort, 2007). This is the reason why DH-based approaches aim at thesaurus and lexicon building, word-sense disambiguation, terminology extraction, models for vocabulary learning, models of semantic priming and semantic deficits, etc.

As we will suggest in the next pages, introducing our methodological framework, verb-centric proposals evolve from Harris' idea. Such verb-based approaches establish that a pivot word, namely belonging to VPs, exists and from its analysis it is possible also to analyse other co-occurring words in a given sentence. It means that in order to understand meaning, we have to deal with sentence contexts, namely linguistic environments[6].

Thus, linguistic environments become a subject to be investigated, also for other disciplines, such as neo-behaviourist psychology in cognitive science. Examining other American-Structuralism works, Lenci (2008), states that

[5]In Chapter IV we define such expressions as Atomic Linguistic Units (ALUs).

[6]These approaches represent the foundations on which semantic-role notions are established. For more information, see Chapter IV.

> *(…) the essence of such models resides in the idea that word meaning depends on the contexts in which words are used, and that at least parts of a word content can be characterized by its contextual representation, to be defined as an abstraction over the linguistic contexts in which a word is encountered.* (Lenci, 2008:9)

Besides, according to the same author, from the '80s, while turning into neo-empiricist paradigms, statistical methods have been applied not only for part-of-speech tagging or syntactic parsing, but also for lexical-semantic analysis (Lenci, 2008:7).

Indeed, among mathematical tools, statistical and stochastic analyses are currently the main means by which contextual features are singled out to describe distributional behaviours (Lenci, 2008).

Actually, the contextual hypothesis maintains that contextual representations may provide more information about contexts and situations in which a word is used. Therefore, these methods of distributional semantics evolve into attempts of analysing also extra-linguistic information.
Lenci (2008) concludes his analysis as follows:

> *Current models of distributional semantics suffer of various shortcomings, but these do not suffice to dismiss the semantic information that can be extracted with distributional analyses, such as those envisaged by Harris". (…) Distributional semantics can be a tool for linguistics to explore these issues and to provide new contributions to cognitive science.*

In our opinion, semantic information retrieval may be achieved just by means of rule-based approaches rather than with statistical/stochastic methods. It means that we have to deal with accurate observation and recording of lexicon and linguistic-item behaviours. However, the techniques introduced in this chapter, can be used to develop a hybrid approach to KP.

2. Machine Learning Techniques

Traditionally, ML deals with all methods and tools allowing structures to automatically learn (i.e. acquire and reuse information) from data processing. In this field, main algorithms are represented by kernel-based techniques (such as Support Vector Machines, Bayes point machines, kernel principal component analysis, and Gaussian processes).
Machine learning is widely used in the following areas:

- Prediction;
- Adaptive behaviour, including control systems, adaptive web-cites, profile management, intelligent agents;
- Pattern recognition, including speech and text recognition, model of user behaviour, etc.;
- Pattern extraction providing human-understandable pattern as an output.

The use of ML techniques in NLP fields is explained by the consideration that most NLP problems can be identified as classification problems (Magerman, 1995; Zavrel *et al.*, 1997).

A wide class of machine-learning algorithms have been developed in order to discover new knowledge, that is to say in order to describe relationships among concepts and their characteristics. Discovery process takes advantage of different techniques of knowledge extraction, e.g. classification and clustering, which are the two most common techniques[7].

Algorithms developed in this area may be classified on the basis of the learning typology they sketch, which may be supervised, semi-supervised, unsupervised, reinforcement.

[7]"Classificatory analysis refers to a set of supervised learning algorithms, which study pre-classified data sets in order to extract rules for classification" (Ahmad & Dey, 2007).

On the other hand, clustering refers to unsupervised learning algorithms, which are used to partition a given set of data elements into homogeneous groups called clusters. "Clustering is one of the principal techniques applied for mining data arising from many fields, some of which are Banking or Medical Informatics information retrieval and bio-informatics" (*Ibidem*).

Supervised Learning consists of an inference process achieved by means of labelled data sets composed of training examples. Each example contains an input object (usually a vector) and an output value (or s*upervisory signal*). A supervised learning algorithm analyses a training data set in order to infer a function, which can be also used to map other examples. If the output is discrete, the inferred function is identified as a classifier, otherwise if the output is continue, it is identified as a regression function. The function aims at predicting the correct output value for any valid input object[8]. Supervised Learning presents several boundaries, and its main issue is represented by the bias–variance trade-off (or dilemma), that is the problem of minimizing in algorithms two sources of error. Such errors, the bias and the variance[9], prevent algorithms to generalize beyond their training set**.**

Unsupervised Learning is especially used in clustering tasks (unsupervised clustering) and in the development of association rule learning.

Semi-supervised Techniques stand for hybrid techniques, which are established on both previous learning processes.

Reinforcement Learning is inspired by behaviourist psychology and is based on observation, which means that the learning process is achieved through trial-and-error interactions with a dynamic environment[10].

[8]https://en.wikipedia.org/wiki/Supervised_learning.

[9]"The bias is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

The variance is error from sensitivity to small fluctuations in the training set. High variance can cause overfitting: modelling the random noise in the training data, rather than the intended outputs".

Source: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff.

[10]In these pages, we do not deal with reinforcement learning, due to the fact that it is closely related to search and planning issues in AI rather than in NLP.

For more information, we suggest the work of Kaelbling *et al.* (1996) available at

https://www.jair.org/media/301/live-301-1562-jair.pdf.

Table 3.1 - Schema of Machine Learning Methods.

Supervised	• Support Vector Machine (SVM) • K-Nearest Neighbours (K-NN) • Decision trees (DTs) • Adaptive Boosting (AdaBoost) • Naïve Bayes (NB) algorithm
Semi-Supervised	• Generative models • Low-density separation • Graph-based methods • Heuristic approaches
Unsupervised	• Clustering k-means • Association rule learning
Reinforcement	• Genetic algorithms and genetic programming • Markov Decision Processes (MDPs) • Adaptive Heuristic Critic • Q-learning • Dyna • …

Another way to classify ML algorithms comes from their belonging to discriminative or generative models.

Discriminative models, also called conditional models, utilise an observed variable *x* in order to model the dependence of an unobserved variable *y*. Such prediction of *y* from *x* is achieved through conditional probability distributions.

On the other hand**, generative models** specify a joint probability distribution over observation and label sequences. Generally speaking, a generative model assumes that the sequence in each class c_i are generated by a model M_{ci}. Such model is defined over some alphabet Σ, and for any string $s \in \Sigma^*$, the model M_{ci} specifies the probability of $P^{Mci}(s|c_i)$ that the given sequence *s* is generated by the model M_{ci}. $P^{Mci}(s|c_i)$ is the likelihood probability that a given sequence fits into a certain class (Al Hasan, 2014).

The difference between such two models is that a generative model is a "full probabilistic model of all variables, whereas a discriminative model provides a model only for the target variable(s) conditional on the observed variables"[11].

[11]Source Wikipedia: https://en.wikipedia.org/wiki/Generative_model.

The large use of ML methods to solve the so-called *linguistic knowledge acquisition bottleneck* is justified by the consideration that, as already stated, most NLP problems can be viewed as classification problems. Indeed, Zavrel sustains that linguistic tasks may be viewed as classification tasks. Orphanos *et al.* (1999) also ascribe linguistic problems to two types of classification: disambiguation and segmentation.
Some examples of disambiguation are:

1. Determine the pronunciation of a letter, given its neighbouring letters;
2. Determine the part-of-speech (POS) of a word with POS ambiguity, given its contextual words;
3. Determine where to attach a prepositional phrase, given a set of other phrases;
4. Determine the contextually appropriate meaning of a polysemic word.

Some examples of segmentation are:

1. Given a letter in a word, determine whether the word can be hyphenated after that letter;
2. Determine if a period is the boundary of two sentences;
3. Determine the boundaries of the constituent phrases in a sentence.

In the following paragraphs, we will present a brief overview of the main ML methods applied mainly in NLP tasks, also providing some research samples in ontology learning. While skipping other methods, we will focus on those ML approaches that have been employed in OL.

3. Supervised Methods

3.1 Support Vector Machine (SVM)

In machine learning, Support Vector Machine (SVM) is included in supervised learning methods. It has been introduced during COLT '92 by Boser, Guyon & Vapnik (1992) and its algorithm is derived from Statistical Learning

Theory, i.e., the Vapnik & Chervonenkis (VC) dimension (Vapnik & Chervonenkis, 1974). Most applications are included in such class of algorithms, and pattern recognition represents one of these. Such methods may be applied to classification or regression problems. Indeed, "SVM is mostly used for classification and regression analysis" (Raikwal & Saxena, 2012). Originally, SVM models were defined for the classification of linearly separable classes of objects.

A SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. In order to classify samples, various distance measures are applied, as for instance the hyperplane, which maximizes the distance from it to the nearest data point on each side.

If such a hyperplane exists, it is defined as the maximum-margin hyperplane which defines the linear classifier as a maximum margin classifier. In Figure 1, we present an SVM trained with samples from two classes in which maximum-margin hyperplane and margins are represented. Samples which are located on the margins are called support vectors.

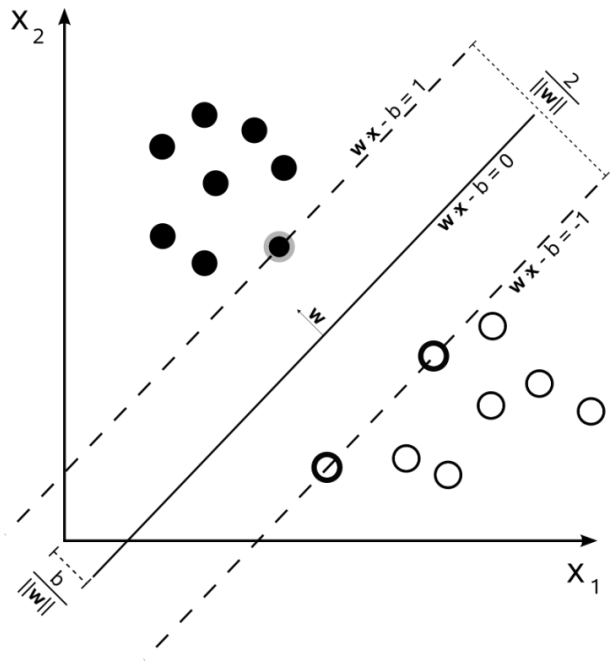


Figure 3.1 - Graphic showing the maximum separating hyperplane and the margin in a SVM[12].

As stated in Todorov (2006), in OL, a SVM provides a measure of concept similarity, which means that it:

- Defines the similarity between concepts by testing the similarities between instances contained in each of the given concepts.
- Uses SVM to create a classifier based on the training examples contained inside one ontology and applies such classifier to classify concepts in another ontology.
- Compares documents relevant to a concept C1 taken from the first ontology to documents relevant to a concept C2 taken from the second ontology.

SVM models may be also utilised as a classification method, for example in the integration of objects from a source taxonomy into a master taxonomy. This is the case of Zhang & Lee's (2004) proposal, which aims at automating the

[12]Image taken from Wikipedia.

https://en.wikipedia.org/wiki/Support_vector_machine.

process by the training of a classifier for each category in the master taxonomy. Subsequently, objects taken from the source taxonomy are classified into these categories. Rather than an inductive one, authors apply a transductive learning, which means a learning suitable to optimize classification on a specific set of samples. Thus, in order to overcome semantic overlap which is a key issue in taxonomy integration tasks they propose a method, called Cluster Shrinkage (CS), based on Trasductive SVMs.

Other tasks which may be achieved by means of SVMs are for example ontology mapping (Todorov, 2006), alignment (Nezhadi *et al.*, 2011) and so on. Actually, SVMs allow to measure concept similarity through joint distributions, variable selection, text categorization and similar techniques.

3.2 K-Nearest Neighbours (K-NN)

K-NN is one of the simplest machine learning algorithms, which has already been used in the beginning of 1970's as a non-parametric technique[13]. K-NN is a classification (or regression) algorithm which combines the classification of the K nearest points in order to determine the classification of a point. It represents a type of instance-based learning, and in pattern recognition or classification is a technique that classifies data using the closest training examples inside problem space. It is supervised because it tries to classify a point on the basis of the known classification of other points. In k-NN, the *k* stands for the number of neighbours which take part in determining a point classification.

In K-NN we may use numeric and symbolic features in order to compute a distance measure. Such a measure specifies the numerical distance between two objects, that is: when two objects are more alike, the distance measure is lower, and, generally, the minimum distance is equal to 0. Measuring distance between objects allows us to group items; in order to obtain such numeric values, we often apply a representation of objects in the form of "feature

[13]Non-parametric techniques apply a statistics not based on parameterized families of probability distributions. Such techniques include both descriptive and inferential statistics. In non-parametric models the number of parameters grows with the amount of training data (Murphy, 2012), such parameters are determined by training data, not by model. For more information, see https://en.wikipedia.org/wiki/Nonparametric_statistics.

vectors". It means that each data object (item) may be represented as an n-dimensional vector, in which dimensions stand for item features (or attributes).

To make concrete example, in Figure 2, we have two classes, A and B, which represent training sets holding five items respectively, yellow and purple dots. We take in account just two features, x_1 and x_2, in order to differentiate classes, which means that the feature-space is 2-dimensional. In order to classify the red star, which is an unlabelled item, as either class A or B, we have to assume that objects belonging to the same class are close one to the other inside the feature-space. On the basis of such an assumption, we can assign a class by a majority vote of the *k* nearest neighbours. If we consider the case *k* = 3, the given item, namely the red star, presents two neighbours from class B and one neighbour from class A; therefore, this leads us to assign the item to class B. If we consider the case *k* = 6, the red star presents two neighbours from class B and six neighbours from class A, and this leads us to assign the item to class A.

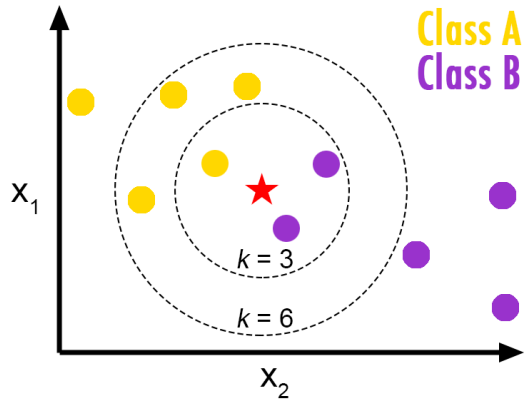


Figure 3.2 – Sample of k-NN classification[14].

Using a vector representation allows us to calculate the distance between pairs of items through standard vector operations, such as cosine of the angle between vectors, Manhattan distance, Euclidean distance, Hamming Distance, and so on.

The use of k-NN provides the following advantages:

[14]Image taken from

http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/.

- Training is very fast
- Complex target functions are easily learnt
- There is no loss of information.

On the other hand, it provides the following disadvantages:

- Query time is slow
- There are lots of storage to achieve
- It may easily be fooled by irrelevant attributes.

Besides, using all training samples in the classification causes a high level of computation complexity. A methodology based on statistical IE method and a K-NN algorithm for developing ontological structures and item classifications is proposed by Shang *et al.* (2006). Therefore, K-NN is usually applied in order to develop methods suitable for ontology-based automatic classifications and ranking of documents (Fang *et al.*, 2007).

3.3 Decision Trees (DTs)

Decision Trees (DTs)[15] have long been considered as one of the most practical and straightforward approaches to classification (Breiman *et al.,* 1984; Quinlan, 1986). Indeed, DTs are classifiers which are expressed as a recursive partition of an instance space and are one of the most popular data mining models.

A tree is a graph *G = (V, E)* in which any two vertices are connected by one and only one simple path. When a vertex is designed as a root, the tree becomes rooted and edges present a natural orientation towards or away from

[15]Usually DTs are applied in both classification and regression tasks. For this reason, they can also fall inside the definitions of classification tree analysis or regression tree analysis. The first term is used when the predicted outcome is the class to which data belongs. On the contrary, regression tree analysis is used when the predicted outcome can be considered a real number. When we refer to both procedures, we use the term Classification and Regression Trees (CART) analysis (see below).

(https://www.academia.edu/5406422/Performance_Analysis_of_Classification_of_Data_using_Structured_Induction_Decision_Tree_Algorithms).

the root. A DT is represented by a rooted tree, in which a vertex is composed by internal, or test, nodes, namely nodes without outgoing edges, and terminal leaf nodes.

Methods that use DTs partite the feature space in a set of regions, then suit a simple model in each region. In such classifiers, the instance space is subdivided into two or more sub-spaces, by each internal node, on the basis of an input value function. Each leaf may contain a class, which represents the most appropriate value, or a vector, which indicates the value probability held by a target attribute. Instance classification is performed proceeding from the tree root down to a leaf, following test outcomes along the path.

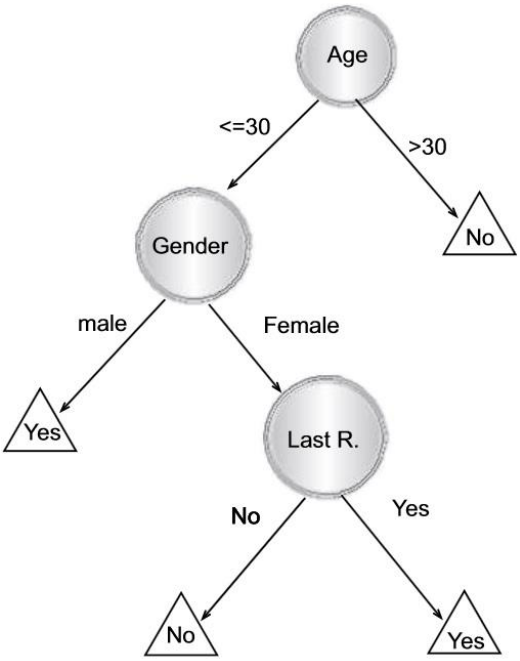


Figure 3.3 - DT Sample[16].

In Figure 3, a DT classifier is presented which shows paths regarding a response to a direct mail by a customer. Circles stand for internal nodes, labelled with the attributes they test; as well, branches are labelled with their values and triangles represent the leaves. Such DT contains both numerical and nominal attributes. Paths along the tree indicate how a potential customer may be answered to by a direct mail, and describe the characteristics of the entire population of potential customers.

[16]Image taken from Rokach & Maimon (2005).

DTs present a certain degree of instability, due to the fact that they are sensible to training data. Indeed, a small change in data may result into a very different set of splits. Such characteristic may go in the detriment of accuracy. Another disadvantage with DTs is that they are prone to overfitting, which means that DT learners may generate over-complex trees not able and/or usable to generalise data adequately. Overfitting may be limited:

- Using specific mechanisms, such as pruning;
- Limiting the minimum number of samples required by a leaf node;
- Setting the maximum depth of the tree.

Before fitting them to DTs, it is recommended to balance datasets, as DT learners might create biased trees if some classes in the datasets are dominating over others.

On the other hand, the main advantage in using DTs is represented by the capability of human interpretation of DT-based classifier. Actually, considering that trees may be visualized, DTs are easily observable as far as the simplicity of understanding and interpretation are concerned. Also, DTs do not require a great deal of effort as for data preparation, while, for example, other techniques necessitate data normalisation. Furthermore, they also are very fast to train and evaluate. Other advantages are that DTs can handle numerical and categorical data and multi-output problems. In such cases, DTs perform appropriately, even if their assumptions are somewhat violated by the model from which the data were generated.

Rokach & Maimon (2005), following Breiman *et al.* (1984), sustain that the accuracy of a DT is directly linked to its complexity. A DT complexity, which is a function coming directly from the stopping of the criteria and the pruning of the methods applied, may be measured according to the total number of leaves, nodes or attributes used, and to tree depth.

DTs may be employed for parsing NL sentences, due to the fact that parsing NL sentences may be interpreted as a sequence of disambiguation decisions. Such decisions concern determining the part-of-speech of words, choosing between possible constituent structures, and selecting labels for the constituents (Magerman, 1995).

DTs seem to be similar to interpoled N-gram models (see Par. 5.2); indeed, both present a robust representational capacity. “The main difference

between the two modelling techniques are how the models are parameterized and how the parameters are estimated" (*Ibidem*).

DTs use a hierarchical tree in order to represent data; thus, each leaf stands for a concept containing a probabilistic description of that concept. The representation of unlabelled data via classification trees is a method used by several algorithms. Nevertheless, most of these are not suitable for clustering large database data (Fisher, 1987). As stated by Rokach & Maimon (2005), this is the case of COBWEB and CLASSIT, an extension of COBWEB.

Clerkin *et al.* (2001) use COBWEB as a concept clustering algorithm for discovering and automatically generating an ontology. Their motivation is traceable in the high pertinence of such an approach as for domains in which expert knowledge lacks.

In the paragraphs which follow, we will introduce some of most common DT-based algorithms.

ID3

ID3 (Iterative Dichotomiser 3) is one of the most simple and oldest DT algorithm, developed by Quinlan in 1986. Such algorithm produces a multiway tree, finding for each node the categorical feature that will yield the largest information gain for categorical targets. With ID3 trees are grown until their maximum size, subsequently a pruning mechanism is applied in order to generalise unseen data.

In Sohn *et al.*(2012), the ID3 algorithm has been applied for generating a domain ontology. Such DT is used to classify the domain web pages and the domain users. Authors' aim is extracting the interests of domain users, while overcoming problems related to such task, which are: defining the user domain and classifying different domain users.

Mirambicka *et al.* (2013) recognize that the main advantage in using ID3 is that such algorithm is robust to noisy and missing data. This characteristic proves to be useful to overcome the issue of missing information, especially when ontologies present a lot of abstract classes. In their work, decision rules, formed by the construction of the DT, are formulated from the root node to the leaf node. "The decision tree modelling helps in gaining a new perspective to the ontology trees and also helps in finding out new relationships between the terms of the respective ontologies" (Mirambicka *et al.*, 2013).

C4.5

C4.5 (Quinlan, 1996) algorithm is an extension of ID3 algorithm. The main difference between these two algorithms is that the restriction on categorical features has been removed. Thus, C4.5 uses a dynamical definition of a discrete attribute, based on numerical variables. Such definition divides continuous attribute values into a discrete set of intervals, which means that continuous attributes, namely coming from an infinite set, may be processed as discrete ones.

Trained trees, e.g. ID3 outputs, are traduced into sets of rules of the "if-then" kind; such rules are evaluated to determine in which order they may be applied.

Elsayed *et al.* (2007) apply C4.5 DT algorithm for discovering and extracting knowledge from structured data. Subsequently, they use the generated DT to build the ontology, represented in XML and OWL languages. According to authors, C4.5 algorithm allows to address issues not covered by ID3, such as avoiding data overfitting, handling continuous attributes and handling training data with missing attribute values. Nodes of DT are mapped to OWL classes, while also tree branches are represented in OWL as classes. Classification rules are represented by leaves in the DT and each rule is exemplified as an instance (or individual) of the class that symbolizes its branch. Authors evaluate their system applying it to two domains: soybean diseases and animal diseases. In the first case, they obtain over 91% of correctly classified instances, with a precision of 99,13%, a recall of 92% and a F-score of 95,43%. This approach differs from the one developed by Wuermli *et al*. (2003), which represents each node as a class and a given tree as hierarchies of classes.

CART

The acronym used for the algorithm CART stands for Classification and Regression Trees. This algorithm was introduced by Breiman *et al.* (1984) and is very similar to C4.5, although CART supports numerical target variables, i.e. regression, and does not compute rule sets. CART manages both categorical and continuous attributes, and also missing values. "Unlike ID3 and C4.5 algorithms, CART produces binary splits" (Lakshmi *et al., 2013*). CART is considered computationally expensive, since during the pruning phase it

produces a sequence of trees from which it is possible to choose the one that minimizes the misclassification rate (Gorea & Buraga, 2006).

CHAID

The acronym CHAID stands for *Chi*-squared Automatic Interaction Detector, originally proposed by Kass (1980). CHAID algorithm produces non-binary trees and it is well suited for the analysis of large datasets. It is particularly popular in marketing research and in market segmentation studies. CHAID is a segmentation technique suitable to analyse categorical dependent variables. Its trees present non-terminal nodes which identify split conditions, "to yield optimum prediction (of continuous dependent or response variables) or classification (for categorical dependent or response variables)"[17]. CHAID works splitting the initial parent group into smaller subgroups, which are considerably different from the dependent variable. This splitting process continues until it is impossible to perform further splits causing the growth of the tree. Actually, a general issue of applying tree classification or regression methods is that the tree may become very large, when the input data are complex.

QUEST

QUEST (Quick, Unbiased, Efficient, Statistical Tree) is a binary-split DT algorithm (Loh & Shih, 1997). Generally speaking, it is faster than CHAID[17], but it is not applicable for regression-type issues (continuous dependent variable, namely when the dependent variable is continuous).

Such method is also called "statistical tree", since it is strongly based on statistical tools for constructing and refining tree. QUEST has been developed as an improvement of FACT[18]. The general idea is (i) to realize algorithms which divide feature selection from the determination of the split, (ii) to convert symbolic features in numeric ones, and (iii) to use statistical tests to make some decision (Grąbczewski, 2014).

[17]Source: http://www.fmi.uni-sofia.bg/fmi/statist/education/textbook/eng/stchaid.html.

[18]FACT (Fast Algorithm for Classification Trees, Loh and Vanichsetakul 1988) is designed to split datasets described by numeric features.

3.4 Boosting and Adaptive Boosting (AdaBoost)

Adaptive Boosting (AdaBoost) is a boosting algorithm[19], formulated in 1996 by Freund & Schapire, which allows to generate a linear classifier out of a set of weak classifiers (Freund & Schapire, 1996 and 1997; Hastie *et al.*, 2001).

AbaBoost allows to work also with classifiers that may come from a continuum of potential classifiers, e.g., as neural networks, linear discriminants, etc. (Rojas, 2009).

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

Figure 3.4 - The boosting algorithm AdaBoost[20].

AdaBoost starts with "given *m* labelled training examples $(x_1, y_1),\ldots,(x_m, y_m)$ where x_i belongs to a domain *X* and the labels y_i are in {-1,+1}. On each round $t = 1,\ldots,T$, a distribution D_t is computed and a given *weak learner* or *weak*

[19]Boosting is a machine learning ensemble meta-algorithm for reducing bias primarily and also variance ("Arcing [Boosting] is more successful than bagging in variance reduction", Breiman, 1996) in supervised learning, and a family of machine learning algorithms which convert weak learners to strong ones ("The term boosting refers to a family of algorithms that are able to convert weak learners to strong learners", Zhou, 2012). From Wikipedia https://en.wikipedia.org/wiki/Boosting_%28machine_learning%29.

[20]Image taken from Schapire (2013).

learning algorithm is applied to find a *weak hypothesis* $h_t : X \rightarrow \{-1,+1\}$, where the aim of the weak learner is to find a weak hypothesis with low weighted error e_t relative to D_t. The *final* or *combined hypothesis H* computes the sign of a weighted combination of weak hypotheses.

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x).$$

Figure 3.5 - AdaBoost Function[21].

This is equivalent to saying that *H* is computed as a weighted majority vote of weak hypotheses h_t, where each is assigned weight a_t" (Schapire, 2013).

AdaBoost has been developed as a minimiser of the upper bound on the empirical error. In other words, during each iteration it reduces the generalization errors, namely training errors.

AdaBoost is very simple to implement and allows both feature selection from very large sets of features and a fairly good generalization. Other advantages are that output meets the logarithm of likelihood ratio and the opportunity of producing a sequence of gradually more complex classifiers. On the other hand, disadvantages come from the suboptimal solution for $\vec{a}$ and potential overfitting in presence of noise (Šochman *et al.,* 2014).

Several variants of AdaBoost have been proposed: Real AdaBoost, Gentle AdaBoost, AdaBoost.M1 and AdaBoost.M2.

4. Unsupervised Learning and Clustering

As previously stated, unsupervised learning is especially used in clustering tasks (unsupervised clustering) and in the development of association rule learning.

Generally speaking clustering is the process through which a data set, constituted by *n* points inserted in a *m*-dimensional space, is partitioned in *k* distinct sets of clusters[22]. Points in cluster are grouped on the basis of

[21]Image taken from Schapire (2013).

[22]We report Äyrämö & Kärkkäinen's cluster definition, that collect it from the clustering literature (Aldenderfer & Blashfield, 1984; Jain & Dubes, 1988):

similarity, which means that such data points are more similar to each other than to data points in other clusters. For this reason, clustering represents one of the principal techniques applied for mining data in many fields.

Thus, several authors have proposed different definitions of cluster analysis. For example, Anderberg (1973) describes cluster analysis aim as to find "*natural groups* from a data set, when little or nothing is known about the category structure".

A more recent description of cluster analysis comes from Jain *et al.* (1999): "the organization of collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity".

From a statistic point of view, Hastie *et al.* (2001) defines cluster analysis as the task of "partition the observations into groups ("clusters") such that the pairwise dissimilarities between those assigned to the same cluster tend to be smaller than those in different clusters".

Thus, cluster analysis is strictly related to investigate the internal structure of a complex data set and to describe it using tools different from the second order statistics techniques (the sample mean and covariance).

According to Ahmad & Dey (2007), clustering process also deals with other three sub-problems:

- *defining a similarity measure to judge the similarity (or distance) between different elements;*
- *implementing an efficient algorithm to discover the clusters of most similar elements in an unsupervised way;*
- *and derive a description that can characterize the elements of a cluster in a succinct manner.*

- "A Cluster is a set of entities which are alike, and entities from different clusters are not alike."
- "A cluster is an aggregation of points in the space such that the distance between two points in the cluster is less than the distance between any point in the cluster and any point not in it."
- "Clusters may be described as connected regions of a multidimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points" (Äyrämö & Kärkkäinen's, 2006).

Due to the absence of a priori knowledge, which indicates data distribution, clustering process becomes more complex than others (*Ibidem*). Also, due to its unsupervised, descriptive and summarizing nature, data clustering has also become a core method of data mining and knowledge discovery (Äyrämö & Kärkkäinen, 2006).

Among clustering algorithms, we may identify two main classes: hierarchical and partitional. The first class, namely hierarchical algorithms, aims at creating a hierarchical sets of clusters. On the other hand, partitional methods, which includes k-mean, bisecting k-mean, k-modes, etc., splits data sets into non-overlapping groups (Duda & Hart, 1973; Jain & Dubes, 1988), trying to minimize clustering error.

Table 3.2 – Clustering-method schema.

Clustering Methods	**Algorithm Types**
Hierarchical Methods	Hierarchical clustering algorithms
Partitional Methods	Error Minimization Algorithms Graph-Theoretic Clustering
Density-based Methods	DTs
Model-based Clustering Methods	Neural Networks
Grid-based Methods	
Soft-Computing Methods	Fuzzy Clustering Evolutionary Approaches to Clustering Simulated Annealing for Clustering

We choose to focus specifically on hierarchical, partitional and soft-computing methods and just to give short descriptions of the other ones.

Density-based methods aim at identifying clusters and their distribution parameters in the information space. The underlying assumption is that points in each cluster may be described by a specific probability distribution (Banfield & Raftery, 1993). The assumption is that data overall distribution is a combination of several distributions. Density-based approaches intend to discover arbitrary shape clusters and not only convex ones. These method are set “to continue growing the given cluster as long as the density (number of objects or data points) in the neighbourhood exceeds some threshold. Namely, the neighbourhood of a given radius has to contain at least a minimum number

of objects. When each cluster is characterized by local mode or maxima of the density function, these methods are called mode-seeking" (Maimon, O., & Rokach, 2005: 335).

Component densities may be multivariate Gaussian (in case of numeric data) or multinominal (in case of nominal data). Therefore, the maximum likelihood principle is one of the proposed solutions to maximize the probability of data. Namely, according to this principle, clustering structures and parameters should be chosen by the maximization of the probability of the data being generated by such clustering structure and parameters.

Parameter estimation may be solved by applying the expectation maximization (EM) algorithm (Dempster et al., 1977), in order to estimate the parameters of the selected distribution. EM algorithm is proposed as a general-purpose maximum likelihood algorithm for missing-data problems. Fraley and Raftery (1998) describe such algorithm as a two-step procedure:

- E-step. Starting from observed data and current parameter estimates, E-step aims at computing the conditional expectation of the complete data likelihood.
- M-step. In this step are maximized parameters usable to maximize expected likelihood from E-step.

Starting from these methods, several algorithms have been developed: for example, the DBSCAN (density-based spatial clustering of applications with noise) algorithm (Ester et al., 1996), AUTOCLASS (Cheeseman and Stutz, 1996), SNOB (Wallace and Dowe, 1994) and MCLUST (Fraley and Raftery, 1998).

Model-based clustering methods also aim at finding characteristic descriptions for each group, which is representative of a concept or a class. Such techniques are used for the optimization of the match between the given data and a mathematical model. Within these inductive methods, DTs and neural networks are the most commonly applied.

Different algorithms are included in **grid-based methods** for clustering. They are all built on the idea of partitioning the space into a finite number of cells forming a grid on which clustering operations are performed. According to Han & Kamber (2001), the main advantage of these methods is represented by fast processing time.

4.1 Hierarchical Clustering Algorithms

On the basis of the criterion applied to partition instances, hierarchical clustering algorithms may be divided into agglomerative or divisive.
Agglomerative clustering, which is based on a top-down approach, merges iteratively two closest clusters, starting from a unique cluster formed by a single item/element. Such merging process is carried on until the accomplishment of a final cluster, containing all data items, or until the achievement of a pre-defined termination condition.

Divisive hierarchical clustering, following a bottom-up process, is based on an inverse path: it starts with a unique cluster, which includes all points, and proceeds splitting such cluster into various cohesive sub-clusters. Splitting process is performed until each point fits in a unique cluster or until a pre-defined termination condition is achieved.

Due to the fact that merging or division may be performed using different similarity measures, hierarchical clustering methods may also be differentiated according to the calculation techniques applied. (Jain *et al.,* 1999) propose the following types:

- **Single-link clustering** measures the minimum, or shorter, distance from each member of one cluster to each member of the second cluster. Such distance represents the distance between this pair of clusters. For such reason, it is also identified as the minimum method or the nearest neighbour method (Sneath and Sokal, 1973)
- On the other hand, **complete-link clustering** calculates the distance measure between two clusters on the basis of the maximum, or longest, distance from each member of one cluster to each member of the second cluster. Therefore, it is also called maximum method or the furthest neighbour method.
- **Average-link clustering** is a method that evaluate the average distance from each member of one cluster to each member of the second cluster. Such value represents the distance between a pairs of clusters. Average-link clustering is also referred to as minimum variance method.

Strengths of hierarchical clustering are versatility and multiple partitions (Maimon & Rokach, 2005). Versatility allows to maintain good performances during data set analysis. On the contrary, multiple partitions, presented as a dendrogram[23], allow to choose among different nested partitions, on the basis of the desired similarity level.

On the contrary, their main disadvantages are represented by the inability to scale well, which means that the time complexity is non-linear with the number of objects, and the absence of back-tracking capability, which means that hierarchical methods can never undo what was done previously (Maimon & Rokach, 2005).

Actually, traditionally, hierarchical clustering algorithms seem not scalable due to their high computational costs, that *de facto* limit their use to not very large database.

4.2 Partitional Clustering Algorithms

Partitional clustering algorithms start from an initial partitioning**,** on the basis of which they relocate instances moving them from a cluster to another. Usually the number of clusters, in which to split data elements, has to be pre-set by the user. Partitional clustering algorithms apply an iterative process to relocate data into pre-defined *k* clusters, minimizing a cost function ζ of the type:

$$\zeta = \sum_{i=1}^{n} \|d_i - C_j\|^q$$

[23]A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering.

Source: https://en.wikipedia.org/wiki/Dendrogram.

where:

- C_j is the center of *j*th cluster and is the center nearest to data object d_i;
- n is the number of elements in data set;
- q is an integer which defines the nature of the distance function (q = 2 for Euclidean distance).

For numeric valued data sets, a cluster centre is represented by the mean value of each attribute, the mean being computed over all objects belonging to the cluster (Ahmad & Dey, 2007).

Various methods are present in such type of cluster analysis, such as error minimizing algorithms and graph-theoretic clustering.

- **Error minimizing algorithms** are most frequently used methods. "The basic idea is to find a clustering structure that minimizes a certain error criterion which measures the "distance" of each instance to its representative value" (Maimon & Rokach, 2005). Usually these algorithms apply the Sum Squared Error (SSE) criterion, measuring the total squared Euclidian distance of instances to their representative values. Within algorithms that employ a squared error criterion, the k-means is one of the most spread (see Paragraph 4.2.1).
- **Graph-theoretic clustering** is a method that produces clusters through graphs. In such representations, each edge connects instances that are represented as nodes. Techniques are different from each other for the way in which the graph is constructed and clusters are detected and separated. One of the most commonly used method is the Minimum Spanning Trees (MST) (Zahn, 1971). In MST edges are weighted in order to identify inconsistent edges, namely those that present a value appreciably different than the average of nearby edge values.

4.2.1 K-Means

K-means, also referred to as Lloyd's algorithm, is a clustering error minimization algorithm that tries to partition a set of points into *k* clusters (c_1, c_2, …, c_k) such that the points in each cluster tend to near each other. This

simplest and most common algorithm, employing a squared error criterion, belongs to unsupervised methods, because the points have no external classification.

The *k* in k-means indicates the number of clusters we want to have in the end. If k = 5, then we will have 5 clusters, or distinct groups, of data elements after we run the algorithm on our dataset.

K-means assumes that documents are real-valued vectors and it clusters points basing on centroids, that is the centre of gravity, or the (arithmetic) mean, of points in a cluster, *c*. Therefore, the mean of all instances in a cluster represents the centre of that cluster.

The formula to calculate the mean to be the centroid of the observations in a cluster is the following one:

$$\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

Figure 3.6 - K-means Algorithm[24].

K-means bases the reassignment of instances to clusters on distance to the current cluster centroids, or one that may equally express it in terms of similarities.

In k-means, given a set of *k* means, which selects a K random documents *{s1, s2, ... sK}* as seeds, the algorithm proceeds alternating two steps:

- **Assignment step:** for each document, d_i algorithm assign d_i to the cluster c_j such that the distance between x_j and s_j - $dist(x_j, s_j)$ - is minimal.
- **Update step**: it means that algorithm updates the seeds to the centroid for each cluster. Indeed, for each cluster c_j we have $s_j = \mu(c_j)$

The algorithm proceeds until it accomplishes a termination function, e.g., a fixed number of iterations, doc partition is unchanged, centroid positions don't change[25].

[24]Image taken from F. Aiolli, University of Padova, course slides. Available at http://www.math.unipd.it/~aiolli/corsi/SI-0607/Lez17.211106.pdf.

Thus, k-means starts with a predefined set of *K* cluster-centres and progresses updating them iteratively in order to reduce the error function.

K-means presents a linear complexity, and such a characteristic differentiates it from other algorithms, e.g. hierarchical clustering methods, which have a non-linear complexity. Dhillon & Modha (2001) indicate other reasons for k-means popularity: its ease of interpretation, simplicity of implementation, speed of convergence and adaptability to sparse data.

On the other hand, one of the boundaries in such algorithm is traceable in the initial partition selection, which is a sensitive task. Actually, the initial selection may make the difference between global and local minimum[26] (Maimon & Rokach, 2005); therefore, the partition is often computed by another algorithm.

Furthermore, k-means is responsive to data noise, which entails the risk of increasing the squared error. Also requiring in advance the number of clusters may be a boundary if no prior knowledge is available. K-means also presents a numeric data limitation, which means that it is frequently limited to numeric attributes. Clustering algorithms, as *k-means* one does, usually apply Euclidean distance measures in order to evaluate distance between items. Various modifications to k-means, such as spherical k-means and k-medoids, have been proposed to allow the use of other distance measures.

For example, Ahmad & Dey (2007) maintain that Euclidean distance measure, used by traditional clustering algorithms, seems inadequate when we deal with element attributes which are categorical or mixed. In order to overcome this issue, authors propose a distance measure able to process mixed data. Due to the spread of large mixed data KB, the inadequateness of traditional distance measures is also considered a weakness by the data mining community. Ahmad & Dey identify two main strategies employed to solve this problem:

---

[25]When the relocation of centres does not reduce the partitioning error, we may suppose that the present partition is locally optimal.

[26]A local minimum, also called a relative minimum, is a minimum within some neighbourhood that need not be (but may be) a global minimum. A global minimum, also known as an absolute minimum, is the smallest overall value of a set, function, etc., over its entire range. Source: http://mathworld.wolfram.com/LocalMinimum.html.

- Converting categorical and nominal attributes in numeric integer values and measuring distances among them to obtain similarity between pairs. Obviously, such solution entails the problem of assigning correct numeric values to categorical attributes, e.g., colour, gender and so on.
- Using discrete quantities to represent numeric attributes in order to employ categorical clustering algorithms. In this process, the risk is represented by information loss.

Thus, Ahmad & Dey propose a variation of k-means algorithm with a modified distance function and a modified definition of the cluster centre. Indeed, as we said, variations within k-means algorithms are traceable in the way they determinate the cluster centre and measure the distance between an object and the centre. Therefore, Ahmad & Dey employ a centre cluster definition similar to the one used for fuzzy clustering[27] and dynamic distance measure.

Although, results seems encouraging, authors recognize the need of other implementations to achieve more optimized performances.

4.3 Soft-Computing Methods

As stated in Das *et al.* (2013), generally speaking, "Soft Computing refers to the science of reasoning, thinking and deduction that recognizes and uses the real world phenomena of grouping, memberships, and classification of various quantities under study"[28]. Usually, in this category various methods are inserted, such as evolutionary approaches for clustering, fuzzy clustering and so on[29]. Evolutionary approaches have been developed in order to solve general optimization problems in clustering. Such methods intend to obtain a globally optimal clustering, using evolutionary operators,

[27]For more information, see El-Sonbaty & Ismail (1998).

[28]The definition of soft-computing clustering is used in opposition to hard-computing clustering. In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster.

[29]Generally, also, Artificial Neural Networks are enclosed in such class; however, we introduce this method in Probabilistic Language Model category (see Paragraph 5.5).

namely selection, recombination and mutation. The process is achieved through encoding candidate clusters as chromosomes.

On the other hand, the introduction of fuzzy clustering is motivated by the need of overcoming overlapping structures in classification tasks.

In fuzzy clustering, each pattern is associated to each clusters to a certain degree, on the basis of a sort of membership function; this means that each cluster is a fuzzy set of all the patterns. This procedure can be defined as a soft clustering schema, opposite to traditional, hard, clustering that associates each instance to one and only one cluster, which lead to the condition of clusters being disjointed from each others. Membership values are utilised to determine the confidence score[30], and namely a larger membership value indicates a higher confidence in the assignment of an instance (or pattern) to a cluster. From a fuzzy partition it is possible to derive a hard clustering, applying a threshold of the membership value (Maimon & Rokach, 2005). Obviously, in fuzzy clustering, designing membership functions represents the main problem: various possibilities may be applied, from similarity decomposition to centroids of clusters. Among these various methods, the fuzzy c-means (FCM) is the most spread fuzzy algorithm, used mainly in pattern recognition. FCM is useful to avoid local minima but it can still converge to the local minima of the squared error criterion (Rokach & Maimon, 2005b). Actually, such a method splits clusters by means of an iterative optimization of the objective function, updating membership and the cluster centres. In other words, this iterative procedure of updating allows to reduce the local minimum, namely the minimum, the result of which depends on the initial choice of membership weights. Otherwise, FCM seems less efficient with the squared error criterion which measures the average of the squares of the errors, namely the difference between the estimator and what is estimated.

[30]Confidence score is metric that describes the relationships between clustering rules and cases. Confidence is the probability that a case described by this rule will actually be assigned to the cluster.

Source:
https://docs.oracle.com/cd/E11882_01/datamine.112/e16808/clustering.htm#DMCON239

5. Probabilistic Language Models and Word Embedding

The goal of language modelling is to build a statistical language model that can estimate the distribution of natural language as accurate as possible. A statistical language model is a probability distribution P(s) over strings S that attempts to reflect how frequently a string S occurs as a sentence.

In order to assign such probability of distribution to sentences and sequences of words, Probabilistic Language Models (LMs) have been developed. Among such models the simplest ones are N-grams, which are used to both estimate the probability of a word, given the previous word, and give probabilities to entire sequences.

Probabilistic LMs may be distinguished into two categories concerning the kind of learning applied: deep or representation.

Deep Learning attempts to learn multiple levels of representation of increasing complexity/abstraction. Also, it aims at obtaining high-quality distributed representations of words, that is word embedding, in the form of continuous vectors. On the other hand, Representation Learning attempts to automatically learn good features or representations[31].

In the last years, deep learning techniques have grown up, increasing interest in developing complex and deep models based on the analysis of large amounts of data, to accomplish different NLP and text mining tasks (Bengio *et al.,* 2003; Collobert & Weston, 2008; Glorot *et al.*, 2011; Socher *et al.,* 2011; Mikolov, 2012; Tur *et al.,* 2012).

In many areas, several researches have applied deep learning methods, for example in language modelling (Mikolov *et al.*, 2013), image recognition (Krizhevsky *et al.*, 2012), sentiment classification (Socher *et al.*, 2013), Speech recognition (Dahl et al., 2012), MNIST hand-written digit recognition (Ciresan *et al.*, 2010).

Deep learning systems usually apply several kinds of multiple-layer neural networks, i.e., Deep Belief Networks (DBNs), Markov Random Fields with multiple layers, etc.

In recent times, word embedding is one of the most spread research area in deep learning. Actually, according to Bian *et al.* (2014):

[31]Source: http://web.stanford.edu/~lmackey/stats306b/doc/stats306b-spring14-lecture15_slides.pdf

While traditional NLP techniques usually represent words as indices in a vocabulary causing no notion of relationship between words, word embeddings learnt by deep learning approaches aim at explicitly encoding many semantic relationships as well as linguistic regularities and patterns into the new embedding space.

Word embeddings is based on the idea of distributed representations for symbols, which means storing information as encoded vectors and spreading encoded information across the entire dimension of the vector (Bengio et al. 2001, 2003; Hinton, 1986). Thus, word embeddings are often seen as a dimensional-vector space where the dimensions can be seen as features potentially describing syntactic or semantic properties. Therefore, as we will see, word embedding may be represented as a parameterised function which maps words in some language to high dimensional-vector space.

In the paragraphs which follow, we introduce some of the main algorithms used in deep learning systems.

5.1 Naïve Bayes (NB) algorithm

The structure of the Naïve Bayes (NB) classifiers, which are the simplest language models (or model-based method), "assumes that the features are independent, so the likelihood probability is simply the multiplication of the likelihood of finding a feature in a sequence given that the sequence belongs to the class c_j" (Al Hasan, 2014).

NB attempts to fit a generative model for documents through training examples and apply model for classifying test examples. On the basis of such NB model, a document is generated first choosing a component $c_j \in C$ according to the prior distribution $P(c_j|0)$ and subsequently selecting a document d_i according the parameters of c_j with distribution of $P(d_i|c_j;0)$. Therefore, the likelihood of a document is given by the following total probability (McCallum & Nigam, 1998):

$$P(d_i|\theta) = \sum_{j=1}^{|\mathcal{C}|} P(c_j|\theta)P(d_i|c_j;\theta)$$

Figure 3.7 - NB probability of document likelihood[32].

Parameters of c_j are estimated on the basis of labelled training documents, which are documents manually annotated with their (correct) class. Given a set of training documents $D = \{d_1, d_2, \ldots, d_m\}$, denoted parameters with *0*, the class prior parameters are calculated as the fraction of training documents in c_j, using maximum likelihood, in which $P(c_j|d_i)$ has a value equal to 1 if *di* belongs to *cj,* and a value equal to 0 otherwise:

$$\hat{\theta}_{c_j} = P(c_j|\hat{\theta}) = \frac{\sum_{i=1}^{|\mathcal{D}|} P(c_j|d_i)}{|\mathcal{D}|}$$

$P(d_i|c_j;0)$ is estimated calculating the posterior probability of each class, given *d*, using Bayes' rule:

$$P(c_j|d;\hat{\theta}) = \frac{P(c_j|\hat{\theta})P(d|c_j;\hat{\theta})}{P(d|\hat{\theta})}$$

Thus, the classifier selects the class with the highest posterior probability. Since $P(d|0)$ is the same for all classes, then *d* can be classified by computing, as follows:

$$c_d = \underset{c_j \in \mathcal{C}}{\operatorname{argmax}}\, P(c_j|\hat{\theta})P(d|c_j;\hat{\theta})$$

Probabilistic LMs compute such probability relying on the Chain Rule of Probability. A general chain rule, or a general product rule, allows the calculation of any member of the joint distribution of a set of random variables using only conditional probabilities[33]. It also used in the study of Bayesian

[32]Figure 10, 11, 12 and 13 taken from McCallum, A. and Nigam, K. A Comparison of Event Models for Naive Bayes Text Classification. In AAAI-98 Workshop on Learning for Text Categorization, Madison, WI, 1998, 41-48.

[33]Source: https://en.wikipedia.org/wiki/Chain_rule_%28probability%29.

networks, to describe a probability distribution in terms of conditional probabilities.

$$\mathrm{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} \mathrm{P}(X_i = x_i \mid X_{i+1} = x_{i+1}, \ldots, X_n = x_n)$$

Figure 3.8 - Chain rule[34].

Figure 3.8 shows the link existing between computing the joint probability of a sequence and computing the conditional probability of a word, given the words by which it is preceded.
When applied to compute joint probability of words inside sentences, the chain rule may be rewritten as follows:

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

Thus, we "could estimate the whole probability of an entire sequence of words by multiplying together a number of conditional probabilities" (Jurafsky & Martin, 2009). Noticeably, due to the creativity of natural languages, we cannot compute the exact probability of a word, given a long sequence of preceding words.

Typically, NB algorithms are applied in OL tasks for concept relation extraction and semantic class labelling of concepts by various scholars.

Among these researches, Agrawal & Srikant (2001) propose an approach to integrate taxonomy developing an NB algorithm, and Sureshkumar & Zayaraz (2015) use an NB classifier in order to extract ontological relations automatically.

5.2 N-grams

N-gram model, preliminary to Hidden Markov Model, is one of the simplest statistical language models. N-gram basic idea estimates the structure of texts, corpora or languages as the probability that different words may occur alone or in sequence, that is their immediate context.

[34]Image taken from Wikipedia.

Thus, n-gram statistics is represented by frequency tables of all previous sets of *n* consecutive words. The *n* indicates the number of words involved in the estimation of (co)occurrence-word probability. N-grams are mainly of two types: unigrams and bigrams.

Unigrams treat words in isolation, while bigrams, more complex than unigrams, take into account some ordering restrictions, i.e., co-occurrence rules, which may occur in languages. Therefore, bigrams consider word immediate context; namely, in a bigram model such context is represented by the preceding word. Even if the structure of n-gram model is simple, calculating the transition probabilities from sample data is a challenge task.

We use a text sample, without normalization[35], to show how a unigram works:

Humpty Dumpty sat on a wall,
Humpty Dumpty had a great fall.

There are twelve words in the text. The token *Humpty*[36] occurs twice, thus it has a probability of 2/12 = 0.166. On the other hand, *wall* occurs only once, thus its probability is 1/12 = 0.083.

On this view, the likelihood of a text is calculated as a function of probabilities of its parts. In other words, if we assume that the choice of each word is independent, then the probability of the whole string is the product of the independent words (Hammond, 2006)[37]. Thus, the probability of the string *Humpty Dumpty* is calculated using single part values: 0.166 x 0.166 = 0.027.

Concerning bigrams, which are a more complex model, it is possible to state that they are suitable to analyse some ordering restrictions in languages or texts.

[35]Text normalization is the process of converting a text in a consistent way before processing it, e.g., removing non-alphanumeric characters or diacritical marks.

[36]It worth to notice that (co)occurrence-word probability is computed without considering ALUs. Thus, n-grams measures co-occurrence of *Humpty,* even if the concept is expressed by means of an ALU, which is *Humpty Dumpty.* For more information, see Chap. V.

[37]http://dingo.sbs.arizona.edu/~hammond/ling178-sp06/mathCh8.pdf.

According to Jurafsky & Martin (2009), "the intuition of N-gram models is that instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words".

Indeed, high-order N-grams calculate the probability that a word occurs as a function of its context.

Such assumption, which means that the probability of a word depends only on the previous word, is known as Markov assumption. In models based on Markov assumption only previous *history* is considered useful to predict word sequences. History includes only last *k* words, which means that the memory used in Markov models is limited, due to the fact that older (i.e. not immediately close) words are considered as less relevant.

Thus, in bigrams the probability that a word may occur is calculated using the word by which it is preceded. Such probability is equal to the co-occurrence of the two tokens, divided by the probability of the preceding token:

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})}$$

Figure 3.9 - Probability in bigrams[38].

Context functions represent the main difference between the bigram model and the chain rule. In the chain rule, the whole context is the probability context; in bigrams, the whole context is the immediate context.

The spread use of N-grams is motivated by the possibility of computing some text and subsequently applying N-grams to generate new text (see Shannon, 1951). New text may be generated by the prediction that a well-formed text should be composed by the repetition of tokens with the higher probability values. Notice that in bigrams such prediction is calculated also computing the probability that some word starts the sentence and that some word ends it. However, even applying such a restriction, the text produced may be not grammatically correct, due to the lack of a correct handling of morph-syntactic rules[39].

[38]Image taken from Wikipedia https://en.wikipedia.org/wiki/Bigram.

[39]We will see in following chapters how a correct formalization of natural languages, which is not based on stochastic methods, may guarantee more accuracy of results.

5.3 Hidden Markov Model

Hidden Markov Model (HMM) is strictly related to *N-grams* and is developed on the basis of Markov chains. In a simple Markov chain, starting from an initial state, chosen randomly, a system evolves in discrete time steps, according to state transition probabilities. Usually a state corresponds to a directly observable quantity. As in FSA, the sequence of steps stands for the string generated/accepted by the machine.

HMM uses a directed graph to picture the state transitions, specifying the initial and final states, in which nodes stand for the set states and arches represent the transition between two states.

Probability values are associated to each arc and all arcs leaving any particular node must exhibit a probability distribution.

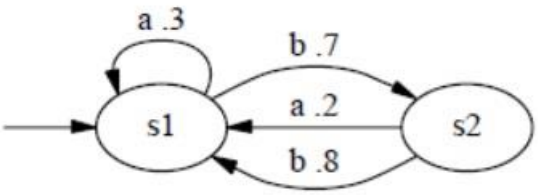


Figure 3.10 – Sample of a Markov chain[40].

The probability that a machine may recognize/generate the appropriate string is computed by means of the product of probabilities, associated to arcs, multiplied together.

Markov chains may be compared to deterministic FSA. The difference between a Markov chain and an FSA is that in the former case probabilities may be associated to each arc. In Markov chains there is only one sequence of states that corresponds to a particular string.

On the other hand, HMMs are definable as a non-deterministic Markov chain, which means that, introducing indeterminacy in the model, a string may be not identified by a unique sequence. Several paths may recognize a single string, e.g. in the model we can introduce multiple start states with different probability values.

[40]Image taken from M. Hammond 2006 Probabilistic Language Models (4/10/06) http://dingo.sbs.arizona.edu/~hammond/ling178-sp06/mathCh8.pdf.

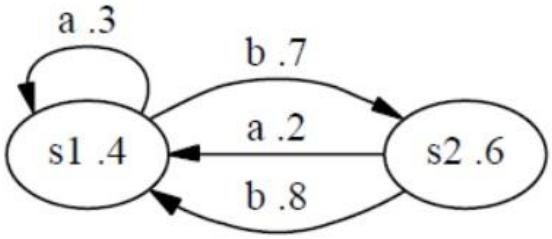


Figure 3.11 – Sample of HMM[41].

In Figure 11, a string *b* could be accepted/generated starting in s_1 and following the arc to s_2, thus the probability is given by 0.4 x 0.7. Also, s_2 may be a start state, in this case the string *b* follows the arc from s_2 to s_1, and its probability is given by the product of 0.6 and 0.8 values. The overall probability of the string *b* is calculated as sum of all possible paths probability. In our example, overall probability of *b* is equal to 0.28 + 0.48 = 0.76.

Another way to produce a non-deterministic model is obtained by the introduction of more arcs for the same state for the same symbol. Indeed, a characteristic of HMM is that they may include both more arcs and various start states.

In a HMM, even if a state produces observable symbols, the state itself is not observable. In other words, the model generates a string in which the initial state, the transition to another state, and the final one are described emitting an output symbol for each state.

HMMs are popular for sequence clustering and classification; they are applied mainly in bioinformatics in *multiple sequence alignment* tasks, which consists in the alignment of a protein sequence to a protein family.

5.4 Probabilistic Context-Free Grammars (PCFGs)

Probabilistic context-free grammars (PCFGs) are context-free grammars in which probability values have been associated to rules. In this way, we are employing a probabilistic model of syntax, by means of which it is possible to develop a statistical parser.

[41]Image taken from M. Hammond 2006 Probabilistic Language Models (4/10/06) http://dingo.sbs.arizona.edu/~hammond/ling178-sp06/mathCh8.pdf.

Thus, if a context-free grammar is formed by a start symbol, a set of terminal and nonterminal symbols and a set of rules, then a PCFG is formed also by rule probabilities.

The probability of a sequence, therefore, is the product of probabilities of all applied rules, and, in case of multiple uses of a rule, the product is calculated factorizing that value as many times as it is used.

For example, we consider a grammar formed by two nouns and two verbs, as the following one:

S → NP VP
VP → V
VP → V NP
V → loves
V → follows
NP → John
NP → Mary

Such grammar produces a parse tree[42] for the sentence *John loves Mary*

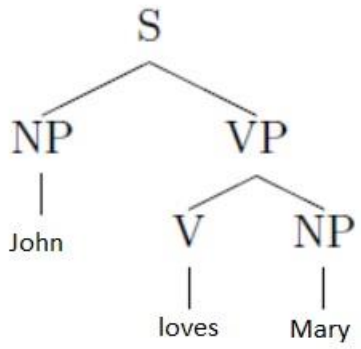


Figure 3.12 - Sample of parser tree.

[42]A parse tree, or parsing tree or syntax tree or derivation tree, is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar.

https://en.wikipedia.org/wiki/Parse_tree.

If we associate each production rule to a probability value, such that any particular nonterminal sum is equal to 1, we convert this context-free grammar in a probabilistic one.

S → NP VP	1
VP → V	0.3
VP → V NP	0.7
V → loves	0.4
V → follows	0.6
NP → John	0.2
NP → Mary	0.8

In our example, *John loves Mary*, the probability is equal to 1 x 0.2 x 0.7 x 0.4 x 0.8 = 0.044.

Obviously, such a method presents some issues when the PCFG is recursive, namely when it generates an infinite number of sentences.

Anyway, such PCFGs allow to probabilistically solve ambiguities and can be easily learnt from treebanks[43].

5.5 Neural Networks Language Model

In the last few years, and mainly in pattern recognition tasks, there has been a growing use of Artificial Neural Networks, also known as Neural Networks (NNs). NNs are computational models based on the observation of human brain neural activities which aim at reproducing human reasoning and meaning processes. Main characteristic of neural networks is to be complex adaptive systems, which means that they adapt their internal structure on the basis of a given processed information flow. In software development, neural networks are used in various tasks, e.g., pattern recognition, time series prediction, signal processing, control, softsensors and anomaly detection.

Basically, the idea of neural networks is based on a perceptron, that is a computational model of a single neuron. A perceptron, invented by Frank Rosenblatt in 1957, is formed by one or more inputs, a processor and a single

[43]Treebanks are parsed text corpora that annotate syntactic or semantic sentence structure. One of the most used is the Penn Treebank. For more information, see Chap. IV.

output. "A perceptron follows the "feed-forward" model, meaning inputs are sent into the neuron, are processed, and result in an output" (Shiffman, 2012).

In Shiffman's description, the workflow of a perceptron may be subdivided into four steps: receive inputs, weight inputs, sum inputs and generate outputs.

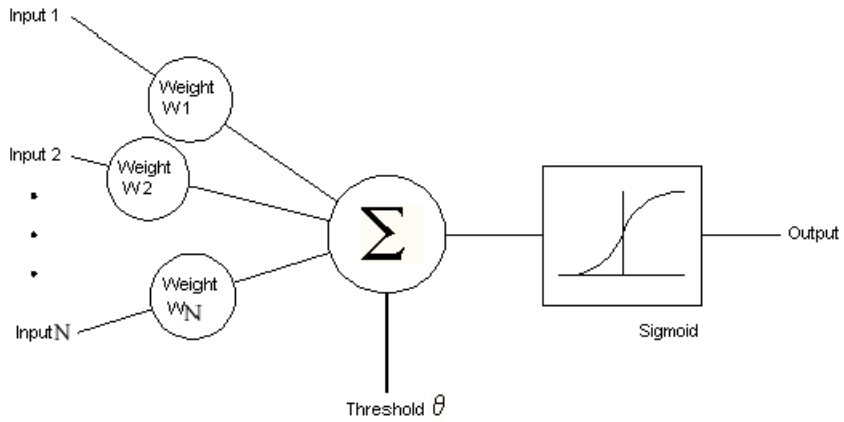


Figure 3.13 - Perceptron Workflow[44].

In 1989, Robert Hecht-Nielsen provided the simplest definition of a neural network: "(…) a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs".

Typically, a neural network is composed of layers, which are made up of a certain number of interconnected nodes containing an activation function[45]. In Figure 14, circular nodes represent artificial neurons and arrows stand for the connection from the output of one neuron to the input of another. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output[46].

[44]Image taken from http://homepages.gold.ac.uk/nikolaev/311perc.htm.

[45]http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html.

[46]Source: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html.

A NN allows to run several logistic regressions[47] at the same time, namely they may run multiple and simultaneous processes to convert continuous inputs/signals into binary outputs. As perceptron, also NNs are feed-forward model based, which means that the network does not contain loops. Networks that allow feedback loops are called Recurrent NNs[48].

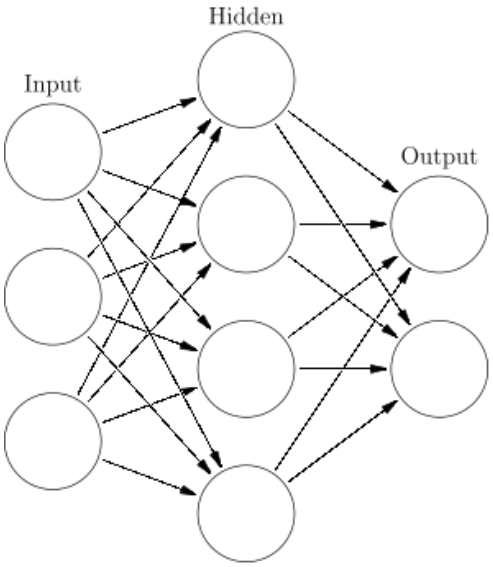


Figure 3.14 - Sample of NN Schema[49].

Since their introduction, NNs have been also used for sequence prediction and language description. In this field, Elman (1990) performed one of the first attempt of applying recurrent NNs in modelling sentences generated by an artificial grammar.

Subsequently, Bengio *et al.* (2003) propose an approach for developing a linguistic model of natural language based on statistical neural networks.
Such and other efforts are justified by the attempts of overcoming N-grams limitations, e.g., the length of pattern representations, or the exponential increasing of possible parameters.

[47]Logistic regression is usually applied in classification tasks, rather than in regression ones. It computes the probability that a set of inputs matches labels.

[48]See https://en.wikipedia.org/wiki/Recurrent_neural_network.

[49]Image taken from https://en.wikipedia.org/wiki/Artificial_neural_network.

In Bengio's model, the inputs are one or more words of language model history, encoded as a one-hot $|V|$-dimensional vector, that is one component of the vector is 1, while the rest are 0, where $|V|$ is the size of the vocabulary. A projection layer, in which input words are included, maps a word vector w_i into a word embedding $v_i \in R^n$ through matrix multiplication, in which also the dimensionality of semantic space is computed. Such word embeddings are concatenated into a single vector, which becomes an input for a standard multi-layer perceptron (Ravuri & Stolcke, 2014).

Furthermore, NNs give the possibility of learning word embeddings. Indeed, neural distribution representations apply neural word embeddings, combining vector space semantics with the prediction of probabilistic models. In these models, words are represented as a dense vector.

The development of a neural network for learning word vectors (Collobert *et al.*, 2011) starts from the idea that a word and its context may be a positive training examples, and on the other hand a random word in the same context may give a negative training sample[50].

Neural network based language models are nowadays among the most successful techniques for statistical language modelling. They can be easily applied in a wide range of tasks, including automatic speech recognition and machine translation, and provide significant improvements over classic backoff n-gram models.

Deep Neural Networks (DNNs) are structured as deep architectures, thus they are able to learn more complex models than shallow ones.

Within neural networks, deep learning ones are the most well-suited to a variegate range of tasks. According to Chintala and Zaremba (2015),

> *Deep learning—neural networks that have several stacked layers of neurons, usually accelerated in computation using GPUs—has seen huge success recently in many fields such as computer vision, speech recognition, and natural language processing, beating the previous state-of-the-art results on a variety of tasks and domains such as language*

[50]http://web.stanford.edu/~lmackey/stats306b/doc/stats306b-spring14-lecture15_slides.pdf.

modelling, translation, speech recognition, and object recognition in images[51]*.*

Recently DNNs are applied in various pattern-recognition tasks, mainly in visual classification problems[52]. Indeed, in the last few years DNNs have been proposed as model for systems in contests on image recognition and classification and similar tasks. In 2012, the contest on visual object detection was won by a DNN system[53].

Also in NLP, DNNs are useful to recognize and learning complex pattern data. As we have seen, indeed, DNNs employ distributed representations of words, such as word embeddings or word vectors. Therefore, vectors may be applied as features in many NLP tasks (Collobert *et al.*, 2011). Word vectors are also useful to capture various linguistic properties (e.g., gender, tense, plurality, semantic concepts)[54]. In their work, Mikolov *et al.* (2013) examine the vector-space word representations, underlying the capability of such representations in capturing syntactic and semantic regularities in language. They gather previous works in the field (Bengio *et al.,* 2003; Schwenk, 2007; Mikolov *et al.,* 2010) to take advantage of the level of generalization that distributed representation achieves. Such level of generalization is not possible with classical n-gram language models. Indeed, n-grams evaluate discrete units without inherent relationship to one other. On the other hand, working with word vectors allows to assign similar vectors to similar words (Mikolov *et al.*, 2013).

DNNs may be trained through both supervised and unsupervised learning. Training a NN language model allows to obtain more than the model itself, namely learnt word representations, usable in other unrelated NLP tasks.

---

[51]Soumith Chintala and Wojciech Zaremba of Facebook AI Research. Understanding Natural Language with Deep Neural Networks Using Torch (2015).

http://devblogs.nvidia.com/parallelforall/understanding-natural-language-deep-neural-networks-using-torch/.

[52]For example, see Nguyen *et al.* (2014).

[53]For example, see Cireşan *et al.* (2013).

[54]Mikolov Tutorial Coling 2014

http://www.coling-2014.org/COLING%202014%20Tutorial-fix%20-%20Tomas%20Mikolov.pdf.

For example Collobert & Weston (2008) and Turian *et al.*, (2010) improve performance in various NLP tasks, employing word representations with complex classifiers. In Semantic Web technologies NNs, and their variants, are used mainly for ontology alignment (Bagheri Hariri *et al., 2006*), ontology learning (Peng, 2010), the construction of a domain ontology (Hourali & Montazer, 2012) and for the discovering of knowledge sources (Caliusco & Stegmayer, 2010).

6. Vector Space Models of Semantics

Vector Space Models (VSMs) of semantics have been developed as attempts to improve the capability of computers to understand the meaning of human language utterances. Recent applications of VSMs are part of deeper semantic technologies. VSM was developed by Salton and others researchers (Salton, 1971; Salton *et al.,* 1975) for the information retrieval system SMART. According to Manning *et al.* (2008) SMART lead the way to modern search engines.

The assumption of VSM represents a document in a collection as a point in a space, namely as a vector in a vector space. In such space, semantic similarity is represented by the closeness among points; thus, points, that are semantically similar, are close together, while other points, that are far apart, are semantically distant. In this approach, also users' queries are considered as points in the vector space, which means that queries are *pseudo-documents* (Turney & Pantel, 2010). VSM-based systems classify documents on the basis of increasing distance from queries; indeed, when distance among points (documents and queries) increases, semantic similarity decreases.

Starting from this hypothesis, various efforts for measuring meaning similarity through concrete algorithms conduct frequently to the development of vectors, matrices, and higher-order tensors.

The main difference between such vectors and VSMs is traceable in the use of element values directly derived from event frequencies, that is the number of times that a given word appears in a given context (Turney & Pantel, 2010). Even if a KB is viewed as a graph, and the graph is represented by an adjacency matrix, the KB is not necessarily a VSM, because values in adjacency matrix may be not derived from frequency values.

As we have seen, in machine learning techniques, classification and clustering usually represent elements as feature vectors. Using vectors in order to represent elements, documents, etc. is a spread technique, even if often values inserted in matrixes are not event frequencies.

We gather Turney & Pantel's proposal, which sustains that VSMs may be classified on the basis of the text frequency matrix they use. Such matrixes may be developed as: (i) term-document, (ii) word-context and (iii) pair-pattern matrixes[55].

Term-document matrix. In a term-document matrix, terms are inserted in the row vectors of the matrix and documents in the column vectors. Therefore, generally speaking a term-document matrix contains a large number of document vectors. "In a term-document matrix, a document vector represents the corresponding document as a bag of words" (Turney & Pantel, 2010). According to Salton *et al.* (1975) the bag of words hypothesis lays the foundation for applying VSMs to IR tasks. The bag of words hypothesis believes that the relevance of documents to a query may be measured representing documents and the query as bags of words. In other words, the relevance of the document to the query may be indicated by the frequencies of words in the document. Therefore, the term-document matrix is useful to catch an aspect of document meaning, namely the one which detects the topic dealt with in the document. Such VSM is one of the first attempt to create an algorithm for extracting semantic information starting from word use.

Word-context matrix. As we have described in Paragraph 1, the distributional hypothesis is related to the idea that words that occur in similar sentence contexts tend to have similar meanings. We mainly refer to Harris' works, but several authors have sustained such hypothesis (i.e., Wittgenstein, 1953; Harris, 1954; Firth, 1957; Deerwester, *et al.*, 1990). Starting from this hypothesis, the VSM has been applied to measuring word similarity. Indeed, Deerwester *et al.* demonstrate how theoretical lines proposed by Harris, Wittgenstein, and Firth could be applied in a practical algorithm.

[55]A text frequency matrix has to be intended as a general structure, while term-document, word-context and pair-pattern are specific cases.

Pair-pattern matrix. A pair-pattern matrix is usually applied for measuring the semantic similarity between word pairs and patterns. “In a pair-pattern matrix, row vectors correspond to pairs of words, such as mason : stone and carpenter : wood, and column vectors correspond to the patterns in which the pairs co-occur, such as \X cuts Y " and \X works with Y" (Turney & Pantel, 2010). This hypothesis is proposed as the extended distributional hypothesis by Lin & Pantel (2001). According to this extended distributional hypothesis, patterns that co-occur with similar pairs tend to have similar meanings. Furthermore, similarity among patterns may be used to infer paraphrases among sentences Lin & Pantel (2001). In opposition to this distributional one, Turney (2008) proposes the latent relation hypothesis, which indicates that pairs of words that co-occur in similar patterns tend to have similar semantic relations. Therefore, while the extended distributional hypothesis sustains that patterns with similar column vectors in matrix tend to present similar meanings, the latent relation hypothesis infers semantic relations from column vectors.

Obviously, other VSMs have been suggested, for example the triple-pattern matrices, which measure semantic similarity among word triples. Other proposals concern the generalization of matrixes, which may be considered as a tensor (Kolda & Bader, 2009; Acar & Yener, 2009).

Anyway, processes based on VSMs may be structured in various phases:

Building the frequency matrix. Generally speaking, a frequency matrix is formed by *events*: that means a certain item which occurs in a certain situation a certain number of times (Turney & Pantel, 2010).

Weighting the Elements. Different ways to accomplish weighting the elements have been proposed, according to the type of matrix used. Such approaches share the idea, developed from information theory, that a surprising event has higher information content than an expected event. Therefore, it is necessary to give more weight to surprising events and less weight to expected events (Turney & Pantel, 2010).

In term-document matrixes, the most spread weighting functions may be ascribed to the Tf-idf (term frequency X inverse document frequency) family (Sparck Jones, 1972).

If we define term frequency, or raw frequency, as the number of times that a term, e.g. a word or a token, occurs in a certain document, then the Tf-idf represents a weighted term frequency. Indeed, Tf-idf assumes that word importance is inversely proportional to the number of times it occurs across all documents. “Although Tf-idf is most commonly used to rank documents by relevance in different text mining tasks, such as page ranking by search engines, it can also be applied to text classification via naive Bayes” (Raschka, 2014).

Other proposals, often integrated with Tf-idf, concern length normalization (Singhal *et al.*, 1996). Indeed, search engines may present a bias in favour of longer documents, if document length is not considered.

Smoothing the Matrix in order to limit the number of vector components. Different techniques are applied improving similarity measurements.

Comparing the Vectors in order to measure the similarity of two frequency vectors.

In this chapter, we have coped with some of the algorithms and theories that are applied in NLP and OL. Generally speaking, the main approach used in order to accomplish OL tasks is stochastic and probabilistic ones, such as NER or relation extraction. However, purely linguistics approaches have been also developed, and in the following pages, we will present our proposal for developing an environment, suitable to accomplish a deep semantic analysis.

IV – ONTOLOGY LEARNING AND POPULATION BY RULE-BASED METHODS

Words are, of course, the most powerful drug used by mankind.
Rudyard Kipling

1. Deep and Shallow Linguistic Processing

In order to analyse natural and historical tongues, linguistic processing applies different kinds of frameworks, also to address the level of the analyses, which may concern - contemporarily or separately - the three different layers of phonology, syntax and semantics. Depending on the depth and granularity of these analyses, two main methods may be identified, which are shallow and deep linguistic processing.

Shallow linguistic processing is used to achieve specific NLP tasks but not to accomplish complex or exhaustive linguistic analyses. Systems based on a shallow approach are generally oriented to tokenization, part-of-speech tagging, chunking, named entity recognition, and shallow sentence parsing. In the last years, the capability of text analysis achieved by shallow techniques has improved, also thanks to the upgrading and updating of such systems. Still, concerning efficiency and robustness, shallow technique results are not comparable to the ones obtainable by means of systems based on more fine-grained analyses.

On the other hand, deep linguistic processing mainly concerns approaches in which linguistic knowledge is applied to analyse natural and historical tongues. Such linguistic knowledge is encoded in a declarative way, for instance in formal grammars and not in algorithms or sample databases. Therefore, formal grammars become the expression of both certain linguistic theory and some operations which are used to check consistency and to define information fusing. For this reason, deep linguistic processing is usually defined as a rule-based approach. By the way this does not mean rule-based approaches are definitely opposite to statistical methods, due to the fact that

statistical methods may be also applied to deep grammars and fine-grained systems.

In deep linguistic processing, rules are used which are based on a linguistic theory driving correct syntax[1] of linguistic entities. Such rules state constraints and apply them to word combinatory mechanisms. As well, words are encoded in a specific lexicon.

Syntax rules are not only related to grammatical correctness, which is useful to define if a sentence is grammatically approved or rejected[2] by the linguistic community who might use it, but such rules may also describe semantic representations. Thus, syntax seems to be able to express both linguistic levels, namely the grammatical level and the meaning one.

Nevertheless, the fact that a sentence is grammatically correct does not imply that it is necessarily meaningful. The most well-known example comes from Chomsky (1957:15):

(1) Colorless green ideas sleep furiously.
(2) Furiously sleep ideas green colorless.

These samples are usually proposed to confirm that grammatical correctness does not imply that a sentence represents a meaning[3]. In Chomsky's words, "the notion 'grammatical' cannot be identified with 'meaningful' or 'significant' in any semantic sense".

Chomsky uses this sentence also to lay the foundation to the introduction of deep structure vs. surface structure concepts (1957), which represent the basis of Transformational-Generative Grammar (TGG). Both deep and surface structures are two levels of representation which are present, hence recognizable, in each sentence (see Par. 2.2).

In the next paragraphs, we will present some of the main linguistic theories from which deep linguistic processing has been developed, together

[1]"One of the chief objectives of syntactic analysis is a compact description of the structure of utterance in the given language". (Harris, 1946:161).

[2]It is worth stressing that as for some grammar checking cases, semantics may be also useful to solve ambiguity.

[3]An interesting discussion is available at

http://www.mit.edu/people/dpolicar/writing/proseDP/text/colorlessIdeas.html.

with those linguistic formalisms applied to NL analysis. Linguistic formalisms and their possible applications are antagonistic to stochastic models, which by the way are widely used, mainly in cognitive sciences. Actually, some boundaries comes from the use of such models in order to infer meaning. Indeed, stochastics approaches analyse words as linear sequences of elements, without taking into account their co-occurrence sentence contexts. On the contrary, linguistic formalisms considers sentences as complex and very often non-semantically compositional systems, the meanings of which cannot be deduced from the sums of the single occurrences/words, but from their combinatory interaction considered as simple systems[4]. All this shows that meanings of a sentence, a phrase or a proposition do not derive from the relationship existing among isolated words, but are the results of the linguistic links existing among words, where linguistic means morph-syntactic.

2. Linguistic Theories

In the following pages, we will introduce some of the main linguistic theories, on the basis of which several methodologies and approaches to NLP and to OL have been developed. We specifically examine two distinct lines: the first one refers to theories dedicated to linguistic analysis, and the second one describes some of linguistic formalisms, namely grammars. Both lines offer different theoretical and practical solutions in order to achieve a language formalization, which may guarantee enhanced proposals for several NLP tasks. In the last part of this chapter, we present the most spread LRs used in linguistic analysis. Such resources may be embedded into NLP tools[5] or used as external knowledge sources.

[4]This is the founding element of Maurice Gross' Lexicon-Grammar, which states that when occurring inside simple (nuclear) sentences, each different use of a lexical unit carries within the specific piece of grammar necessary to provide a precise meaning.

[5]We give a wide definition of NLP tools, which to us include any environment usable to perform and achieve one or more NLP tasks. Therefore, among such linguistic tools, we enumerate:

- Rule-based parser
- Heuristic patterns
- Dependency analysis
- dependency treebanks and dependency parsing

2.1 Harris and the Distributional Theory

As we have seen in the previous paragraphs, the Distributional Theory is largely employed also to describe word distributions in vector spaces. Such approach is directly derived from Harris' works. "The work of Zellig S. Harris in language, grammar, and information, and in the methodology of linguistics, is remarkable for its consistency and integrity over a span of almost 60 years, culminating in an elegant and comprehensive theory of language and information"[6]. Among his acknowledgements, he is accredited as the founder of the distributional methodology, summarized in his works (Harris, 1946 and 1951). The central idea of the distributional methodology is that meaning may be inferred by means of the analysis of distributional information. This implies that words occurring in similar contexts are semantically similar; thus, if we describe sentence forms, we may also describe sentence meanings. Therefore, following the structuralist linguistic ideas of Bloomfield, Harris considers semantics encompassed in grammar. For this reason, it is worth dealing with descriptive Linguistics in order to define distributional relations of words within sentence contexts.

In *From Morpheme to Utterance* (1946), Harris investigate linear distributional relations of phonemes and morphemes broadening "the technique of substitution from single morphemes (e.g. man) to sequences of morphemes (e.g. intense young man)", assuming that when we deal with Descriptive Linguistics (DL), substitution becomes an essential mechanism:

> *(…) we take a form A in an environment C-D and then substitute another form B in the place of A. If, after such substitution, we still have an expression which occurs in the language concerned, i.e. if not only CAD but also CBD occurs, we say that A and B are members of the same substitution-class, or that both A and B fill the position C-D, or the like.*

- Parsing
- Syntactic parsing
- Semantic parsing
- Tokenization
- Tagging POS
- Finite-State-Transducers for Natural Language Processing.

[6]Source: http://zelligharris.org/description.html.

Thus, among those words occurring inside an utterance, Harris identifies a pivot word, called *operator*, which requires (i.e. selects, or better "attracts") the occurrence of one or more words, called *arguments*.

Operator-argument relationship may be described as the frequency, or likelihood, evaluation of a word occurrence, concerning a pre-defined occurrence number of a certain operator. Such evaluation is not measured in random sentences, but in couple or triad of words which are in an operator-argument relationship.

Indeed, analysing word combinations which may occur in a language, Harris (1988) identifies three constraints able to preclude such combinations: partial-order, likelihood and reduction.

- The partial-order constraints creates sentence structures, namely it provides a partial order of words in a sentence, defining grammatical relations. Indeed, according to Harris it "is (roughly) an ordering in which some words are higher or lower on some scale than others, while some are neither higher nor lower than others" (Harris, 1988:10). Thus, such constraint limits word combinations, due to the fact that for each word it identifies its arguments (word classes). Therefore, the given word may occur in a sentence with zero or more arguments, establishing a dependence relation with a partial order. For example, for the verb *eat*, the words *man*, *apple* are in the classes of its arguments, but there is not the pair *man, walk.* A word with zero argument is called zero-level word, while a word with nonzero argument is defined as an operator on that argument. There are three levels of required words: (I) at least one zero-level argument, (II) at least one first-level operator, (III) at least second-level operator.

  When the constraint is satisfied, it means that when in a word sequence all source words presents their requirements satisfied, then there is a sentence. In this way, the partial-order creates the sentence structure. In Harris work, the partial-order states that "in the argument position next to a given word operator, the frequency (or probability) of certain words – those not in the argument class for that operator – is zero" (Harris, 1998:13).

  Therefore, the partial-order establishes the dependence of word on the dependence property of words, not just on a specified class of words.

Harris acknowledges that the operator-argument relation, generated by such dependence relation, has some correspondences with the functors used for categorial grammar in Logics (see Paragraph 3.3).

- The likelihood constraint concerns the choose of a particular word for a sentence and it aims at describing the mechanism through which some combinations are more likely than others. With such constraint Harris deals with the concept of semantic expansion, due to the fact that the likelihood allows a word to increase its meaning and to have different meanings in different operator-argument environments.

  Not all words have equal frequency in respect to their operator or argument; thus, likelihood under an operator, or over an argument, indicates the probability per fixed number of occurrences of such operator (or argument). In other words, we can estimate the occurring frequency of a word as the likelihood that word has of being in the position for its argument.

  "The set of words having this higher-than-average likelihood is called the selection (…). The central meaning of a words is given by (the meaning of) the selection of arguments under it or of operator over it" (Harris, 1988:17).

  Harris also underlines that there exist words which present an exceptionally high likelihood, namely words which may be accepted as argument pretty much for every operator (i.e. *someone*, *something*). On the other hand, there are also words with an exceptionally low likelihood in particular situations.

- The third constraint refers to the specification of types of reduction, even to zero, "in the phonemic shape of particular word occurrences" (Harris, 1988:20). The reduction is applicable to material with high likelihood; it means that are reducible words which present an exceptionally high likelihood in a given position. Harris attributes low information to these words, recognizing an inverse relationship between likelihood and information. Thus, the possibility of operating a reduction is generated by both high likelihood and low information; or in other words, in a particular environment an exceptionally low likelihood, together with a high information, restrict reductions[7]. The

[7]Such statement of inverse relationship between likelihood and information is also retrievable in the development of term frequency for inverse document frequency (TF-IDF).

> reduction process operates only on word shape and visibility and it does not change the partial order of the involved words.
>
> As stated in Johnson (Nevin & Johnson, 2002:144), describing the reduction constraint, Harris "switches to a process model in which words progressively 'enter' into the structure". Indeed, reductions may be explained as ordered rules applied as words enter; namely operators and arguments hold the informational conditions, under which a reduction may happen.
>
> In order to show how reduction constraint works together with the two previous constraints, we quote a specific Harris' sample:
>
> *To see that the reduction applies not to a word as such but to a word occurrence in a high-likelihood position, note that in colloquial English, where going to can be reduced to* gonna*, we can find* I'm gonna make it *from* I am going to make it*, but not* I'm gonna the next room*. The reason is that before nouns,* going to *is at selectional frequency only before certain ones of them (*New York, the next room*, but not before* word *or* time*); but before operators,* going to *is at selectional frequency before all of them (*go**,*** make it, speak up, *etc.). Hence* going to *has total high frequency only before operators, and is reducible only there* (Harris, 1988:23).

Furthermore informational contribution, hold by operators and arguments, entails that the lexicon has a central role in the theory. Such a lexicon centricity is immediately referable to the Lexicon-Grammar (LG) (Gross, 1984) framework, developed on the basis of the transformational theory of Harris (1968) and widespread before operator-grammar formulation (see Paragraph 2.4).

We may conclude that in Harris' Operator Grammar, language is described in terms of word combinations, while meaning may be represented in a statistical model through a symbolic approach.

2.2 Transformational-Generative Grammar (TGG)

Generative Grammar originates in the work of Chomsky and his associates, developed since the 1950s.

As stated in Lightfoot's introduction to Syntactic Structures, we may identify three phases at work on Generative Grammar. The first phase starts

with *Syntactic Structures* (1957) and *Aspects of the theory of syntax* (1965), in which Chomsky deals with the expressive power of grammars, in order to cope with different levels of representation (1957) and lexicon (1965). The second phase culminates in Government and Binding models, centred on the power of derivations, in order to produce the very general operations and principles of the theory of (Universal) Grammar.

The third and final phase is traceable in *The Minimalist Program* (1995) in which Chomsky revises the whole TGG framework by means of economy principles.

Chomsky declares his aim in the first pages of Syntactic Structures:

> *The ultimate outcome of these investigations*[8] *should be a theory of linguistic structure in which the descriptive devices utilized in particular grammars are presented and studied abstractly, with no specific reference to particular languages. One function of this theory is to provide a general method for selecting a grammar for each language, given a corpus of sentence of these languages* (Chomsky, 1995:11)

As for lexicon, the Projection Principle is stated in *Knowledge of Language: Its Nature, Origin and Use* (1986), where Chomsky declares that each "lexical structure must be represented categorically at every syntactic level" (Chomsky 1986:84)

Besides, concerning the independence of grammar, Chomsky is convinced that "the fundamental aim in the linguistic analysis of a language L is to separate the grammatical sequences which are the sentences of L from the ungrammatical sequences which are not sentences of L and to study the structure of the grammatical sequences. The grammar of L will thus be a device that generates all of the grammatical sequences of L and none of the ungrammatical ones" (Chomsky, 1957:13). Also, "(…) each grammar is related to the corpus of sentences in the language it describes in a way fixed in advance for all grammars by a given linguistic theory" (Chomsky, 1957:14).

And then "I think that we are forced to conclude that grammar is autonomous and independent of meaning, and that probabilistic models give

[8]Investigations finalized to determine the underlying properties of successful grammars (Note from the editor).

no particular insight into some of the basic problems of syntactic structure" (Chomsky, 1957:17). Therefore "(…) we argue that this relation[9] can only be studied after the syntactic structure has been determined on independent grounds" (*ibidem*, note 4).

Besides, Chomsky and others argue that phrase-structure grammars are not "adequate for giving a full grammatical description of sentences in English" (Harman, 1963:597). According to Herman, such idea, based on a particular definition of phrase-structure grammar, may be modified slightly in order to avoid transformational rules into generative grammars.

According to Chomsky, a transformational process is based on a transformational grammar which works with some transformational rules, namely adding, deleting, moving or substituting of words.

Generally speaking, in any sentence structure there are two elements: an NP and a VP. Chomsky also identifies a Deep Structure (D-Structure) and a Surface Structure (S-Structure): the first one refers to meaning, bringing the semantic component, the second one concerns the phonological component.

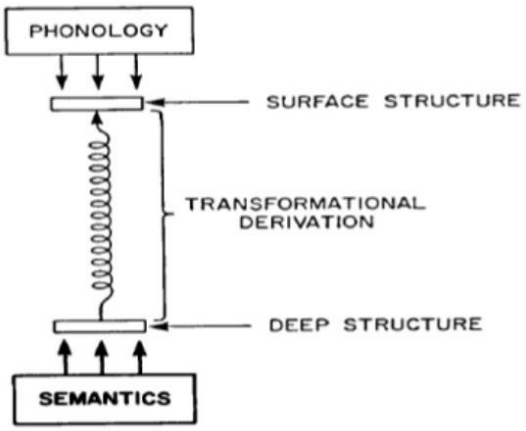


Figure 4.1 – Aspects of transformational grammar model[10].

Thus, Chomsky supposes that there exists a deep structure of language, namely the syntactic base, and a surface structure, which stands for the phonological level. The syntactic base is formed by a series of phrase-structure

[9]The relation between semantics and syntax (Note from the editor).

[10]Image taken from Chomsky (1965).

rewrite rules, i.e., a series of (possibly universal) rules that generates the underlying phrase-structure of a sentence, and a series of rules (called transformations) that act upon the phrase-structure to form more complex sentences.
Therefore, Chosmky's model of representation presents three main features:

1. The deep structure, which determines the meaning of a sentence
2. The surface structure, which defines the pronunciation, or phonetic interpretation
3. Transformations, which allow to convert the semantic level into the phonetical one.

All languages have the same deep structure, but they differ from each other in surface structures, because of the application of different rules for transformations, pronunciation, and word insertion.

D-Structure is represented using hierarchical tree diagrams, or phrase structure trees, which describe grammatical relationships between words and phrases inside a sentence. On the other hand, the formal rule system specifies in which way deep structures have to be transformed into surface structures.

Another important distinction made in TGG is the difference between language competence (the subconscious control of a linguistic system) and language performance (the speaker's actual use of language).

Language competence stands for the knowledge speakers have about their own native tongue. On the contrary, language performance is the way(s) in which speakers actuate such knowledge to form and understand grammatical sentences in their own native tongue.

Although the first work in TGG was mainly focused on syntax, later studies have applied the theory to the phonological and semantic components of language.

In *The Minimalist Program* (1995), Chomsky revises the phrase structure concept acknowledging an important role also to lexicon, which is in a relationship with syntax[11] in a way very similar to the one described by Harris

[11]In *The Minimalist Program*, Chomsky also revises the X-bar Theory, that he first proposed and that further Ray Jackendoff (1977) developed, introducing the Bare Phrase Structure (BPS).

http://web.mit.edu/norvin/www/24.902/phrasestructure2.html.

and Gross. Indeed, he assume that "syntax provides three fundamental levels of representation, each constituting an 'interface' of the grammatical system with some other system of the mind/brain: D-Structure, Phonetic Form (FP), and Logical Form (LF)". Thus, he declares that "the level of D-Structure is directly associated with the lexicon" (Chomsy, 1995:131).

He also develops ideas encompassing economy of derivation, i.e.: transformations occur just to match interpretable features with uninterpretable ones; and economy of representations, i.e. sentence structure complexity.

Usually the term TGG is applied to indicate contemporary works in Chomsky's Revised Extended Standard Theory (REST) and Government Binding (GB; Chomsky, 1981), but not, for example, GPSG, LFG or Arc-Pair Grammar (APG).

2.3 Tesnière and the Valency Theory

Tesnière is acknowledged as the father of dependency, which is one of the main streams of today's structuralistic syntactic theory and also of modern Dependency Grammars (DGs). The concept of dependency may be observed in the works of various grammarians, and under different forms; however, Tesnière is the first to introduce direct word-word dependencies. Such dependencies may be described using tree representations (stemmas), which suggest the analysis of syntactic structures and show the verb centrality in Tesnière's theory. Thus, due to the fact that the verb assumes a central role, it is placed at the root of all syntactic structures.

In *Éléments de syntaxe structural* (1959), "Tesnière rejected much of the terminology of syntax that preceded him, declaring that morphologists had imposed their nomenclature on the study of syntax and thus confused our understanding of syntax (ch. 15)" (Osborne, 2013:263).

> *La croyance dans le caractère morphologique de la syntaxe est à tel pont ancrée dans l'esprit de Meillet*[12] *et de la plupart de ses élèves, qu'ils ont été jusqu'à s'approprier purement et simplement la teminologie*

[12]In this chapter, Tesnière refers to the work of A. Melliet *Linguistique historique et linguistique générale* (Note from the editor).

> *syntaxique, sans même se rendre compte qu'ils dérobaient ainsi aux syntacticiens un bien qui leur appartenait essentiellement, et privés duquel il luer devient bien difficile de traiter de syntaxe d'une façon vraiment syntaxique, puisque les morphologistes se sont ingéniés à qui mieux à donner à tous les termes syntaxiques une signification morphologique* (Tesnière, 1959:35) [13].

Therefore, he introduced new terms to describe syntax, and despite the fact that these have not become standard, the impact of Tesnière's work has not been reduced by this shortcoming.

Furthermore, Tesnière does not employ the term *grammaire de la dépandence,* since he does not differentiate dependency and constituency (Osborne, 2013)[14], even if he use both concepts. Sentence is considered as an *ensemble organisé* (organized set) in which *mots* (words) are *constituants* (constituent elements). A word in a sentence is not isolated as in the dictionary, due to the fact that we can perceive *connexions* (connections) between such word and its neighbours. The totality of these connections among words represents the structure of a sentence. Furthermore, structural connections establish dependency relations among words, joining a superior term and an inferior term. Thus, the superior term is called *régissant* (governing word) and the inferior term is named *subordonné* (subordinate) [15].

[13]The belief in the morphological character of syntax is such an anchored bridge in the spirit of Meillet and of most of his students, that they decided to take the definite ownership of syntactic terminology without even realizing that they were stealing to syntacticians a property that essentially belonged to them, deprived of which it become difficult for them to process syntax in a really syntactic way, since morphologists have striven to discover who of them was better in assigning a morphological significance to any syntactic term. (Translation by the editor).

[14]As underlined by Osborne (2013), such difference is introduced later in the reception of Tesnière work. According to Jurafsky and Martin (2009:489), David Hays (1964) may have been the first to employ the term dependency grammar.

[15]Translation by the editor.

[1] Le connexions structurales établissent entre les mots des rapports de dépendance. Chaque connexion unit en principe un terme supérieur à un terme inférieur.

[2] Le terme supérieur reçoit le nom de régissant. Le terme inférieur reçoit le nom de subordonné (p.13 ch.2).

In his analysis of simple sentence (Fr. p*hrase)*[16] Tesnière distinguishes three elements: verb, actants, and circumstantial complements.

> *[1] Le noued verbal, que l'on trouve au centre de la plupart de nos langues européenes, exprime tout un petit drame. Comme un drame en effect, il comporte obligatoirement un procès, et le plus souvent des acteurs et des circonstances.*
>
> *[2] Transposés du plan de la réalité dramatique sur celui de la syntaxe structurale, le procés, les acteurs et les circonstances deviennent respectivement le verbe, les actants et les circonstants* (ch. 48 p.102) *[17]*.

Therefore, the verb expresses the process, actants are the participants in the process, while circumstantial complements convey the circumstances, namely time, place, and so on, in which the process happens.

Tesnière admits clearly that he does not share the logical opposition between subject and predicate, on which traditional grammar theories have laid their foundations. Indeed, traditional grammars, from Aristotle's one to that of Port-Royal, apply such an opposition to describe sentence context schema. Tesnière considers the opposition between subject and predicate as not pertinent to linguistic aspects (*faits de langue*). Indeed, he believes that this logical approach is related to an opposition among concepts, which is not traceable in a sentence such as: *filius amat patrem* (ch. 49, p.104). In this sample, the opposition between subject and predicate is not marked by an opposition among words. Indeed, *amat* sticks together the predicate element, *ama-*, and the subject element, *-t*. While, there exists an opposition among elements which compose the subject, *filius…-t,* and the predicate, *ama-…patrem*. According to Tesnière, such intricacy among elements does not comply with the logical opposition between subject and predicate, while it is perfectly supported by the verb centrality. Furthermore, the subject and the

[16]Tesnière uses the French term *phrase* to indicate a sentence.

[17][1] The verbal junction, which is at the centre of most of our European languages, expresses a little drama. Being a concrete one, this drama necessarily includes a trial, and usually actors and circumstances.

[2] Transposed from the plane of the dramatic reality to the one of structural syntax, the trial, the actors and the circumstances respectively become the verb, the actants and the participants. (Translation by the editor).

predicate cannot be put into comparison, due to the fact that the subject is often expressed through just one word or not completely expressed. On the other hand, the predicate frequently holds more elements than the subject does and, in some cases, nature and internal structure of such elements are entirely similar to subject ones. In such cases, subjects and predicates have to be located on the same level, in order to underline the interchangeability among actants. Locating subject and predicate on the same level is possible just if the verb has a central role. Indeed, the hypothesis, which assumes that a VP has a central role in the sentence schema, allows us to establish a symmetry between two NPs. For example, the sentence *My best friend loves your young sister* expresses a symmetry between the first NP *my best friend* and the second NP *your young sister.* Thus, assuming *loves* as the central node in our sentence, we may describe NPs in stemmas underlying that such actants may be interchangeable.

Rejecting the opposition between subject and predicate, Tesnière lays the foundation for the valency theory which involves the verb and its actants. The verb is considered as a hooked atom, which is able to attract a variable number of actants according to the number of its hooks. Thus, the valence of a verb is founded on the number of its hooks and consequently on the number of attracted actants (ch. 97, p.238). It is worth noticing that not all valences of a verb have to be employed, and that there are cases in which some valences rest unused. The valency assumed by a verb varies within limits between zero, i.e. impersonal verbs, and four, i.e. verbs of possession transfer.

As we have already stated, Tesnière's work is considered as the starting point of dependency grammar theories. Such theoretical tradition "comprises a large and fairly diverse family of grammatical theories and formalisms that share certain basic assumptions about syntactic structure, in particular the assumption that a syntactic structure consists of lexical elements linked by binary asymmetrical relations called dependencies. Thus, the common formal property of dependency structures, as compared to representations based on constituency, is the lack of phrasal nodes" (Nivre, 2005)[18].

[18]It is worth to underline that Tesnière does not distinguish between dependency and constituency, even if he employed constituency in his theory of *transfer* (Fr. translation).

2.4 Lexicon-Grammar Framework

As we have seen, many theories have been developed starting from Harris' concepts of Operator-Argument (1982) and transformational rules (1964).

Lexicon-Grammar (LG) framework combines Harris' structuralist, transformational and distributional deductions with the notion of morpheme and the method of commutation, or equivalence, among different morphemes, proposed by Bloomfield (1933). Initially LG, proposed by Maurice Gross (Gross, 1986b and 1989), during the '60s, was set up for French, then developed for and applied to Italian by Elia, Martinelli & D'Agostino (EMDA, 1981; Elia, 1984). Actually, such framework is also applied to describe and analyse several languages (e.g. Portuguese, Spanish; English, German, Norwegian; Polish, Czech, Russian, Bulgarian, Croatian; Greek, Arabic, Korean; Malagasy; Chinese; Thai). Such descriptive methodology may achieve efficient results also in automatic NL analysis and parsing, through specific lingwares[19] developed according to this framework and completely dedicated to NLP.

LG represents an empirical approach, due to the fact that it aims to obtain a recording of linguistic data starting from the observation of linguistic phenomena. This means that LG does not apply a hypothetical reasoning, but it achieves the empirical observations[20] of linguistic *acts* evaluated in their concrete contexts of production and usage.

[19]Lingwares are applications related to natural language processing. Among lingwares developed on the basis of LG framework we may mention UNITEX, Cataloga and NooJ.

UNITEX (http://www-igm.univ-mlv.fr/~unitex/) is a corpus processing system, based on automata-oriented technology. Its concept was born at LADL (Laboratoire d'Automatique Documentaire et Linguistique).

Cataloga is a software built by Annibale Elia, Alberto Postiglione and Mario Monteleone (Elia, Postiglione & Monteleone, 2010).

NooJ will be presented in the following chapter, as the main tool applied in our environment.

[20]Joseph Harold Greenberg was the first who introduced the empirical method in linguistic studies, also originating what today is known as Corpus Lingustics (CL). He founded his works on quantitative data taken both from a single language or from a wide range of languages. "According with him, following an empirical and functionalist method means to found researches on a sample of languages as wide as possible. On the contrary, a logical-deductive and rationalist method, such as Chomsky's one, founds researches on

As Gross states, "accumulating data is obviously not an aim in itself. But in all natural sciences it is a fundamental activity, a necessary condition for evaluating the generality of phenomena" (Gross, 1979, p. 866).

LG, directly derived from mathematical language models (Harris, 1982; Schützenberger in Gross *et al.*, 1973), aims at formalizing any mechanism of word combinations in order to describe syntax and word behaviours in sentence contexts.

Indeed, LG considers lexicon as a group of terminal values, in a formal grammar of natural languages, which have to be associated to ordered sequences on the basis of independent combinatory behaviours and rules. Thus, lexicon is not separable from syntax, namely every lexical element, occurring in a sentence context, holds a grammatical function which combines with grammatical functions of other constituents. Combinatory behaviours are driven by co-occurrency and restriction-selection rules.

(1) The *Parliament* discusses investment laws (Parliament=: human noun)
(2) The *Parliament* is empty (Parliament =: locative noun).

In the previous samples, we notice that *parliament* may have two meanings, according to co-occurrence and restriction-selection rules. In (1) *parliament* indicates a human noun, that means an assemblage of persons representing the supreme legislative body of a state. On the other hand, in (2) *parliament* refers to a locative noun, namely a building or room in which members of Parliament work.

Such samples clarify the necessity to achieve a formal description of natural languages, which has to be exhaustive and complete. An exhaustive and complete description may be accomplished just through an accurate observation of lexical entries and their combinatory behaviours, which allows to account for both syntax and lexicon. These empirical observations, accomplished by native-speaker linguists, provide data which are formalized in LG LRs.

---

the properties of a single tongue" (Marano, 2012:52). The difference between the empiricist approach and the rationalist one is delineated by Greenberg, which prefers the first one. Applying such method, he also proposes a set of Linguistic Universals based primarily on a set of 30 languages (Greenberg, 1963).

As all empirical investigation model, also LG identifies a base context from which to start observation. Actually, LG considers simple sentence as the minimal operative linguistic meaning contexts, which means that in such contexts we may evaluate word behaviours, in terms of co-occurrences, selection restrictions and distributions. "More specifically, a simple sentence is a context formed by a unique predicative element (a verb, but also a name or an adjective) and all the necessary arguments selected by the same predicate in order to obtain an acceptable and grammatical sentence"[21] (Marano, 2012:50).

LG theoretical framework was firstly announced as directly derived from the failure of Generative Grammar, a topic which is clearly recalled by Gross in his work with the same title (Gross, 1979). Indeed, Gross identifies his failed attempt to construct a transformational-generative grammar of French using Chomsky's approach as the motivation on which he proposes a new method.

Gross provides various samples, which seem not to be adequately considered in GG analysis, for demonstrating the need of accumulating consistent data. Observing and recording linguistic phenomena is essential for the development of a theoretical hypothesis able to "go further than the description of data taken from a high school grammar" (Gross, 1979:868).

Generative Grammar considers sentences just regarding the formalism, namely the importance of a linguistic example is evaluated just for confirming a theory rather than another.

Furthermore, Gross also underlines the importance of diachronic analysis for describing properties of linguistic phenomena in order to distinguish general properties from accidental ones.

"In syntax, the fundamental type of experiment consists in constructing and evaluating sequences of words whose structure varies with three basic combinatorial deformations: permutation, insertion, deletion" (Gross, 1979:870).

Gross cites Lakatos (1978) to define a grammar as "a model of morph-syntactic knowledge acquired by native speakers and to criticize generative syntax". Indeed, he affirms that generative syntax purpose seems to be just an abstract representation of sentences artificially created for such aim. Generative syntax models are mainly inclined to describe local situations

[21]Further information about simple sentence may be retrieved in Gross (1968).

without considering non-local constraints. Just an extensive and exhaustive classification of lexical items and their local constraints makes us able to articulate statements about syntactic rules in a given language. Even if Gross recognizes the importance of formal grammars, he judges negatively the fact that "linguists have not directed their efforts at building and studying particular grammars, but at looking for abstract constraints on whole classes of grammars" (Gross, 1979:882).

Due to these assumptions, LG draws upon a manually based methodology, which also represents the main source for the development of LRs, useful in NLP applications.

In chapter V, we will propose an in-depth analysis of the LRs, namely electronic dictionaries and local grammar, which are applied by LG to describe any natural language.

3. Grammar Formalisms

Grammar formalisms represent languages which may be used to describe languages in themselves, namely to describe:

- The set of sentences which are encompassed by that language (i.e., the string set)
- The structural properties of such sentences (i.e., their syntax)
- The meaning of such sentences (i.e., their semantics)

In other words, this means that a grammar formalisms amounts to a *metalanguage*, namely a descriptive tool. The usage of such metalanguage is motivated by different reasons, mainly the need:

1. To have a tool usable to describe natural languages;
2. To delimit the class of possible natural languages;
3. To specify a machine-readable and interpretable representation of natural languages.

Each metalanguage, chosen to be used in code writing, characterizes important parameters in grammar formalisms:

- *Linguistic felicity: The degree to which descriptions of linguistic phenomena can be stated directly (or indirectly) as linguists would wish to state them.*
- *Expressiveness: Which class of analyses can be stated at all.*
- *Computational effectiveness: Whether there exist computational devices for interpreting the grammars expressed in the formalism and, if they do exist, what computational limitations inhere in them* (Shieber, 1986:3).

Expressiveness has a central role in such grammar formalisms, as for instance in Generalized Phrase Structure Grammar (GPSG) and Lexical Functional Grammar (LGF), which deal with formal linguistic universals.

From '80s, as for grammar formalisms, also different approaches have been proposed to transformational grammar, mainly as alternatives one to the other. Such approaches have been developed to describe syntactic theories; also, they are used to identify a general framework called Unification Grammar, which include LFG, GPSG, HPSG and CG. Unification grammars aim at analysing languages through the description of static constraints on information associated to structured expressions, while GGT works on transformations of the expressions themselves (Sag *et al.*, 1986). In other words, unification-based descriptions of language consider constraints merely as elements which add information, without performing structural changes, namely all linguistic constraints are monotonic.
As stated by Sag *et al.*,

> *In such theories the linguistic objects under study are associated with linguistic information about objects, in which information is modelled by mathematical objects called Feature Structures. Linguistic phenomena are modelled by constraints of equality over the feature structures; the fundamental operation upon the feature structures, allowing solution of such systems of equations, is a simple merging of their information content called Unification* (1986:238).

Thus, the assessment of common threads shows the tendency to merge these theories under a general framework, even if they present differences, which are significant in some cases.

Beyond the differences, such methods describe linguistic phenomena applying stating constraints as equality conditions over partial information structures. According to Sag *et al.* (1986), such systems, being monotonic, present some advantages over derivational methods. In the following pages, we will analyse some specific grammar formalisms which are encompassed in the framework of unification grammars.

3.1 Dependency Grammar Formalisms and the Meaning Text Theory

In this paragraph, we will introduce dependency grammars, which derive from Tesnière's work. We may identify two main streams of theories: the first one is based on Tesnière's structural syntax, the second one holds constraint-based theories of dependency grammar.
According to Nivre (2005),

> *among these* (those derived from Tesnière work) *we find Word Grammar (WG) (Hudson, 1984, 1990), Functional Generative Description (FGD) (Sgall et al., 1986), Dependency Unification Grammar (DUG) (Hellwig, 1986), Meaning-Text Theory (MTT) (Mel'čuk & Polguere, 1987), and Lexicase (Starosta, 1988). In addition, constraint-based theories of dependency grammar have a strong tradition, represented by Constraint Dependency Grammar (CDG) (Maruyama, 1990; Harper and Helzerman, 1995; Menzel and Schröder, 1998) and its descendant Weighted Constraint Dependency Grammar (WCDG) (Schröder, 2002), Functional Dependency Grammar (FDG) (Tapanainen and Jarvinen, 1997; Jarvinen and Tapanainen, 1998), largely developed from Constraint Grammar (CG) (Karlsson, 1990; Karlsson et al., 1995), and finally Topological Dependency Grammar (TDG) (Duchier and Debusmann, 2001), subsequently evolved into Extensible Dependency Grammar (XDG) (Debusmann et al., 2004)*.

In addition, we will rapidly cope with the Meaning Text Theory, referring to Kruijff's framework (2001) of Dependency Grammar Logic (DGL) for a synthesis of dependency grammar and categorial grammar.

The most spread Dependency Grammar Formalisms are:

Word Grammar

WG (Hudson, 1990) is based on general graphs instead of trees. The ordering of two linked words is specified together with their dependency relation, and extraction of, e.g., objects is analysed by establishing an additional dependency called visitor between the verb and the extractee. Hence, WG does not cleanly separate dependencies from word order (Bröker, 1998).

Functional Generative Description

Sgall *et al.* (1986) assume the existence of a language-independent underlying order, represented as a projective dependency tree, mapped via ordering rules to the concrete surface realization. This theory is multistratal, and it distinguishes five levels of representation, which are phonological, morphematical, morphonological, analytical (surface syntax) and tectogrammatical (deep syntax)[22].

Dependency Unification Grammar

DUG (Hellwig, 1986) defines a tree-like data structure for the representation of syntactic analyses. The theory is non-projective and handles surface order using positional features. By these, also partial orderings and discontinuities can be handled.

Functional Dependency Grammar

FDG (Jarvinen & Tapanainen, 1997) distinguishes between dependency rules and rules for surface linearization. It follows Tesniere's model not only in being non-projective but also by adopting Tesniere's notion of nuclei. Nuclei are the primitive elements of FDG structures, possibly consisting of multiple lexemes.

Broker (1998).

Surface order and dependency structures constitute two separate pieces of information. Broker links structurally dissimilar word order domain

[22]This linguistic framework continues the tradition of Prague School, focusing on the phenomenon of the so-called topic-focus articulation.

structures to dependency trees to achieve a lexicalized, declarative and formally precise natural language description (Debusmann, 2000).

Meaning Text Theory

The Meaning Text Theory (MTT) aims at representing a correspondence between meaning and text, making explicit rules able to describe such correspondence. In order to do this, MTT employs a dependency-based approach and considers syntactic information encompassing into the lexicon, as most of the contemporary linguistic theories do. As Mel'čuk & Polguère (1987) state:

> *This theory puts strong emphasis on the development of highly structured lexica. Computational linguistics does of course recognize the importance of the lexicon in language processing. However, MTT probably goes further in this direction than various well-known approaches within computational linguistics; it assigns to the lexicon a central place, so that the rest of linguistic description is supposed to pivot around the lexicon*.

MTT is interested in strictly linguistic meaning, which means that the literal meaning of utterances is achievable just on the basis of linguistic knowledge, without reference to extra-linguistic contexts. Thus, (set of) meanings and (set of) texts are considered as formal objects which may be described through a formal language. Such sets are both infinite and characterized by (a) finite (set of) relationships, which create a correspondence among elements of meaning and text sets, namely formal rules. The formalization of these relationships, which create a correspondence between meaning and text, is expressed as a many-to-many rule. A many-to-many rule entails the presence of synonymy, namely one meaning expressed by many texts; and of ambiguity, namely one text which expresses several meanings.

In MTT, meaning is considered as the invariant of synonymous paraphrases, that is the principle on which also WordNet synsets are developed (see Paragraph 4.1). Due to the fact that it is part of speaker knowledge and language, meaning belongs to linguistic data, therefore it is accessible to speakers.

As for MTT, Mel'čuk 's formalism assumes seven strata of representation of an utterance[23]:

1. The Sem(antic) R(epresentation), namely the meaning.
2. The D(eep-)Synt(actic) R(epresentation).
3. The S(urface-)Synt(actic) R(epresentation).
4. The D(eep-)Morph(ological) R(epresentation).
5. The S(urface-)Morph(ological) R(epresentation).
6. The D(eep-)Phon(etic) R(epresentation), or phonological representation.
7. The S(urface-)Phon(etic) R(epresentation), or phonetic representation proper, namely the text.

As remind also in Kahane:

> *many contemporary theories assume syntactic and morphological levels. The particularity of MTT is to consider them as intermediate levels between the semantic level (the meaning) and the phonetic level (the text). Thus, the correspondence between meanings and texts is completely modular: a correspondence between the semantic and deep-syntactic levels, a correspondence between the deep-syntactic and surface-syntactic levels, a correspondence between the surface-syntactic and deep-morphological levels, etc.* (2003:4).

Furthermore, there exist six corresponding elements, which are the transition between two adjacent levels, *n* and *n+1*:

1. The Semantic Component
2. The Deep-Syntactic Component
3. The Surface-Syntactic Component
4. The Deep-Morphological Component
5. The Surface-Morphological Component
6. The Deep-Phonetic Component.

[23]Although the term utterance seems to be almost vague, Mel'čuk and Polguère (1987) consider the sentence as their basic analysis unit. Nevertheless, MTT is not limited to sentences, dealing with sequences of sentences.

Each component has the same internal structure based on three types of rules:

a. A well-formedness rule, which guarantees a correct representation of source level;
b. A well-formedness rule, which guarantees a correct representation of target level;
c. And transition rules proper, which govern the application of transition from a level, *n,* to another, *n+1*.

In such representation model, the difference between Surface- and Deep- sublevels is justified by the presence of text-related and meaning-related phenomena.

To justify the structuring of such a model, Mel'čuk and Polguère give the following logical description:

> *To sum up, the synthesis of a sentence appears in the Meaning-Text framework as a series of subsequent transitions, or translations, from one representation to the next one, beginning with SemR; the analysis takes of course the opposite direction, starting with the SPhonR or with the written text* (1987:264).

In order to manage such a representation model, MTT has to be meticulously based on lexicon, which is codified in a specific format, i.e. Explanatory Combinatorial Dictionary (ECD). ECD is composed by lexicographic units which include lexical items, a word or a set phrase, "taken in one well-specified sense". A dictionary entry constitutes a *lexeme*, or *phraseme*, described accurately according three *zones*: the semantic zone, the syntactic zone and the lexical co-occurrence zone.

"MTT uses rules for mapping unordered dependency trees of surface-syntactic representations on to the annotated lexeme sequences of deep-morphological representations. Discontinuities are accounted for by global ordering rules" (Nivre, 2005).

3.2 Generalized Phrase Structure Grammar (GPSG)

"The project of explaining constraints on observed grammars as arising in part from grammar formalisms of low expressive power was the impulse behind Generalized Phrase Structure Grammar (GPSG, Gazdar 1981; Gazdar

et al. 1985), which tried to capture as much as possible within a strictly context-free formalism" (Steedman & Baldridge 2011).

Starting from the basic assumption of Generative Grammar, which states that languages may be considered as collections of expressions in the language itself, Gazdar *et al.* (1985) refer to the works of Montague (1970) and Brame (1981) to assume "that the grammars of natural languages should define not merely the expressions corresponding to sentences, but also subsentential expressions of all categories. (…) An interpreted formal system defining the membership of the collection of linguistic expressions, and assigning a structure and interpretation to each member, is required" (Gazdar *et al.*, 1985:1).

GPSG belongs to the class of unification-based grammars and may be defined as an alternative to GGT. GPSG has only one level of syntactic description, at the opposite of Chomsky's two level-based one, and therefore it does not present transformations.

As stated by the authors, the main goal of GPSG is to delineate a constrained metalanguage, able to define the grammars of natural languages. Such universalism has to be intended as entirely represented in the formal system and not expressed by statements made in it. This means that if a feature is universal, it becomes a consequence of the grammatical language itself. For example, when we deal with a feature which presents some values, such as *finite*, we know that such feature implies being verbal and non-nominal. In other words, we know that only verbs may present tense, thus we know that the feature refers to a verbal form. In a grammar theory, this statement may be set up as a universal feature of co-occurrence restriction. Instead, in GPSG, such universals are built into the metalanguage, since authors declare that one of the goals of this theory is "the construction of theories of structure of sentences under which significant properties of grammars and languages fall out as theorems, as opposed to being stipulated as axioms" (Gazdar *et al.*, 1985:5).

Similarly, semantic rules are not given by the grammar, which means that a semantic rule is not necessarily stipulated for each rule in the syntax. Instead, a universal mapping from syntactic rules to semantic translations is supposed as existing.

Indeed, authors declare that:

> *The semantic type assigned to any lexical item introduced in a rule (e.g. the lexical information that possess denotes a function from noun phrases denotations to verb phrase denotations); and the syntactic form of the rule itself are sufficient to fully determine (i) the form of the lexical translation rule, and thus (ii) the set of logical expressions which can represent the constituent defined by the syntactic rule, and thus (iii) the model-theoretic interpretations of that constituent* (Gazdar *et al.*, 1985:8).

Such claim seems to be different from Montague's statement, which asserts that in an NL grammar each syntactic rule is connected to a semantic rule. This semantic rule defines the meaning of a constituent, while its form is specified by the syntactic rule[24].

Starting from these assumptions, GPSG develops a theory of features, identifying two kinds of rules: atom-valued and category-valued ones. Atom-valued features are structured as Boolean values and apply symbols such as the following ones:

(1) [-INF] → which indicates *finite*, an inflected verb as *loves*
[-INV] → that stands for *inverted,* namely a subject-auxiliary inversion, as for instance *Is John loved?*
[+INF] → which indicates *infinitival*, such as *to love*

On the other hand, in category-valued features the value is represented as a nonterminal symbol, which is itself a feature specification. Among category-valued features, SUBCAT represents a feature that identifies the complement of the verb, while SLASH indicates missing constituents.

If we consider a transitive verb phrase, VP, then VP/NP, or VP[SLASH = NP], stands for a VP when an NP is missing.

(1) Max **hit** the floor → hit [sb/sth] → VP
(2) Who did Max **hit**? → hit [e] → VP/NP

[24]Sometimes this statement is called *rule-to-rule hypothesis* (Bach, 1976).

In order to manage sentences like (2), also called *wh-questions,* it is necessary to add another feature besides SLASH; thus, we may encode wh-questions using +WH feature.
Such feature allows us to differentiate phrases like the following ones:

(3) -WH[Max]
(4) +WH[who]

When features do not contradict each other, it may be applied a unification mechanism, which is similar to the set union operation.

A further component of GPSG is represented by metarules, which are comparable to transformations in transformational grammar, and generate related phrase structure rules. For example, starting from an active sentence, i.e. Max eats the apple, a passive metarule allows to define rules for generating the passive sentence directly, marking VP with the [+PASSIVE] feature (atom-valued).

Due to the fact that metarules capture generalizations, which are results of local transformation in a transformational grammar, we are able to manage long-distance dependencies (Horácek *et al.*, 2011).

In the late '80s, interest in GPSG is decreased due to some formal issues and to the opinion that natural languages should be considered as mildly context-sensitive[25]. In spite of this, many concepts and ideas formulated within the GPSG framework (ID/LP format, head feature convention) have been incorporated by Head-Driven Phrase Structure Grammar.

[25]In computational linguistics, the term *mildly context-sensitive grammar formalisms* refers to several grammar formalisms that have been developed with the ambition to provide adequate descriptions of the syntactic structure of natural language.

Every mildly context-sensitive grammar formalism defines a class of mildly context-sensitive grammars (the grammars that can be specified in the formalism), and therefore also a class of mildly context-sensitive languages (the formal languages generated by the grammars).

For more information:

http://en.wikipedia.org/wiki/Mildly_context-sensitive_grammar_formalism.

3.3 Combinatory Categorial Grammar (CCG)

Combinatory Categorial Grammar (CCG) is a linguistically expressive formalism, firstly proposed by Steedman & Baldridge (2011) and developed starting from Categorial Grammar (CG) (Ajdukiewicz, 1935; Bar-Hillel, 1953). CG, one of the oldest lexicalized grammar formalism, identifies all grammatical constituents using a syntactic type. A syntactic type can be used to recognize constituents as "a function from arguments of one type to results of another, or as an argument. Such types, or categories, are transparently related to the semantic type of the linguistic expression itself, differing mainly in the inclusion of information about language-specific linear order" (Steedman & Baldridge, 2011).

CCG also represents a variety of lexicalized grammars, in which the syntactic type of inputs drives applied syntactic rules in order to parse NL. This means that syntactic types, also called categories, allow to classify a constituent as either a primitive category or a function. Primitive categories hold tags as N, NP, PP, S, etc., and they may be distinguished by further features, such as number, case, inflection, etc. Functions, for example verbs, accept categories such as VP, namely the type of their results and that of their arguments. Furthermore, such categories also state the order for argument combinations, and decide if they have to occur to the right or the left of the functor.

The principle of Categorial Type Transparency (Steedman, 2000) is applied to syntactic categories. Such principle states that:

> *For a given language, the semantic type of the interpretation together with a number of language specific directional parameter settings uniquely determines the syntactic category of a category*.

The main difference between CCG and CG is that CG (Ajdukiewicz, 1935; Bar-Hillel, 1953) explores syntactic combinations applying functional rules just to the arguments on the right and left of a function. Thus, functions and primitive categories are considered exclusively as a simple and ordered combination of the function itself and of the arguments that are selected by the function. On the other hand, CCG proposes further rules to describe most different ways in which categories combines. As they give the name to the theory, such rules are called combinatorial rules and present the set of

characteristics driven from syntactic types. In addition, those rules present the semantic correspondences they have with the combinatory elements identified by Curry and Feys (1958). Mainly, these rules express few basic operations, which are related to some directionally specialized instantiations, e.g. *type-raising,* functional *composition* and *substitution* (Steedman & Baldridge, 2011). On the basis of these combinatorial rules, it is possible to identify functions and to specify the kinds and directionality of their arguments. For example, in a sentence we may identify a NP and a VP. Assuming that the VP is composed by a Transitive Verb (TV) and another NP, we can associate some specific lexical entries to the TV.

S → NP, VP
VP → TV, NP
TV → {*love*, *eat*, . . .}

Love := (S\NP)/NP

Thus, syntactic information derived from context-free production rules are transferred to lexical entries. This means that *love* is identified "as is" only under certain assumptions which can be retrieved from the context, and that are:

1. A rightward-combining functor over a domain, represented as /NP,
2. A leftward-combining functor, represented as S\NP.

Furthermore, in their turn both S\NP and /NP may represent function categories.

Basing on Mark Steedman's CCG formalism, an open source NLP library written in Java has been proposed, that is OpenCCG, the OpenNLP CCG Library[26]. This library applies multi-modal extensions, described in Baldridge (2002), to CCG. According to the author, lexicon representation and computational processing may be improved by incorporating devices and category constructors from related categorial frameworks, namely Multi-Modal CCG.

[26]http://openccg.sourceforge.net/.

Concluding, due to the introduction of composition and type raising, CCG applies a surface structure which is more adaptable than the traditional notions of surface constituents. In such surface structure "most contiguous substrings of a grammatical sentence are potential constituents, complete with a compositional semantic interpretation, for the purposes of the application of grammatical rules" (Steedman & Baldridge, 2011:49).

3.4 Head-driven Phrase Structure Grammar (HPSG)

Head-driven phrase structure grammar (HPSG) is influenced by GPSG[27] and is characterized by theoretical richness, formal rigor and computational versatility.

It is a lexically based theory of phrase structures, and its paradigm was founded by Pollard & Sag (1994), starting from an earlier exploratory work (Pollard & Sag, 1987). The main HPSG goal is to create a model that describe how human language is structured in the mind, as the LFG theory also does (Carnie, 2013). Its name is derived from the role of grammatical heads with associated complements; such a role drives HSPG approach. In Linguistics, heads are represented by words or phrases, which apply syntactic and semantic restrictions on other phrases, namely complements. For instance, in a NP the noun is the head, in a VP the verb, and so on (Proudian & Pollard, 1985).

According to Levine & Meurers (2006), HPSG approach presents two main components:

- A representation of grammatical categories, which is explicit and highly structured. Such representation is encoded as typed feature structures and its complexity is justified by theoretical backgrounds and empirical considerations[28].

[27]According to Fliekinger *et al.* (1985), the main variation between HPSG and GPSG is realized by the repositioning of linguistic information. Indeed, linguistic information are inserted into the lexicon, not in phrase structure rules.

[28]"The theory of an HPSG grammar is a set of description language statements, often referred to as the constraints. The theory essentially singles out a subset of the objects declared in the signature, namely those which are grammatical. A linguistic object is admissible with respect to a theory if it satisfies each of the descriptions in the theory and so does each of its substructures" (*Ibidem*).

- A set of rules standing for descriptive constraints, which are applied on grammatical categories. The set of descriptive constraints conveys linguistic generalizations and selects those expressions which are recognized as valid elements in the NL. From a linguistic point of view, HPSG descriptive constraints are formed by: "a) a lexicon licensing basic words, b) lexical rules licensing derived words, c) immediate dominance schemata licensing constituent structure, d) linear precedence statements constraining constituent order, and e) a set of grammatical principles expressing generalizations about linguistic objects" (*Ibidem*).

HPSG defines descriptive constraints as features that represent the main tool in its linguistic description. Such features are essentially attribute-value pairs, which may be easily represented through attribute-value matrixes (AVMs) in which feature structures are inserted. Thus, an AVM contains the entity, in our case the linguistic object, its features, or attributes, and their value.

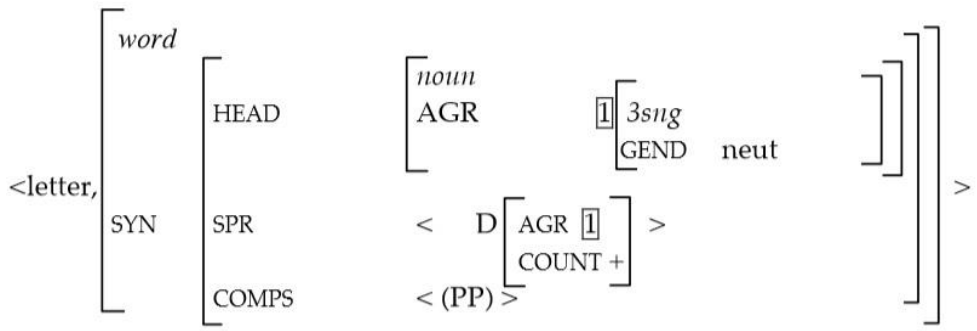


Figure 4.2 - Sample of a lexical entry description in AVMs[29].

In Figure 2, the linguistic object *letter* is represented with its attributes, which may hold either atomic (simple) or complex values. For example, an atomic value is represented by *neut*(er), which stands for the value of the GEND(er) attribute. On the other hand, a complex value may be composed again by a complex AVM, i.e., AGR(eement) value that is formed by another attribute GEND(er) and a simple value *3sng*.

[29]Image taken from Sag and Wasow (1999:132).

HPSG framework is employed in several researches; for instance, the CSLI Linguistic Grammars Online (LinGO) Lab at Stanford University[30] develops linguistically accurate grammars using this formalism.

3.5 Lexical Functional Grammar (LFG)

Lexical Functional Grammar (LFG), introduced by Joan Bresnan in 1982, is a formalism developed for expressing generalizations about syntax of human languages.

LFG states that there are two levels which exist simultaneously: a constituent structure, also called *c-structure*, and a functional structure, also called *f-structure*. The first one, the c-structure, stands for the syntactic representation level, while the f-structure incorporates information from the *c-structure* and the lexicon. Thus, LFG supposes that c-structure differs across languages, because syntax varies across languages. On the other hand, f-structure is considered universal, because it contains all information which allow semantic analysis of a sentence.

Such a two-level structure seems apparently very similar to the one proposed by Chomsky; however, LFG differs from TGG, since it rejects transformational assumptions about syntax. In spite of this, LFG shares some goals with TGG, which leads to assume that LFG "is therefore a variety of generative grammar, an alternative to transformational theory" (Falk, 2001). Indeed, LFG also asserts that, in a sentence, words are structured in constituents and such constituents may be represented in a tree structure, with the c-structure, generated by rules. However, in opposition to TGG, LFG does not use transformations; therefore, it does not present a D-structure. Structure rules produce directly the c-structure, which corresponds to the S-structure in TGG, and, thus, LFG may process displaced items in other ways. In LFG, the c-structure may be not defined by a tree, because grammatical functions are considered primitive notions, which means that they may be not derived in some way (Carnie, 2013). "Every sentence has an f-structure that represents grammatical functions. In the f-structure, a particular NP will be identified as being the subject of the sentence, quite independent of the tree structure associated with the sentence" (Carnie, 2013:3). In other words, the f-

[30]http://lingo.stanford.edu/.

structure is composed by the set of all the attribute value pairs for a sentence. Such information, which realize the f-structure, are stored in the lexicon, namely in the lexical entries which are co-occurring in the sentence. In fact, syntactic behaviours are described for each lexical entry. This means that a verb entry reports information about predicate, form and arguments whit which it deals. For example, the verb *eat* requires two obligatory arguments: the subject and the object.

To show which shape may have an f-structure, we propose here one of Carnie's samples, specifically the one for the sentence *the professor loves phonology*:

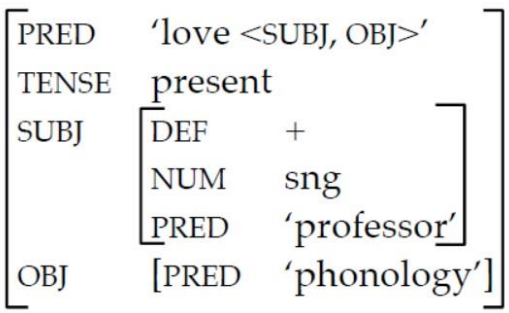


Figure 4.3 - LFG f-structure sample[31].

SUBJ and OBJ are formed by submatrices in which information about their internal structure are contained. C-structure and f-structure are interlinked using variables, as showed in Figure 4.3.

[31]Image taken from Carnie (2013).

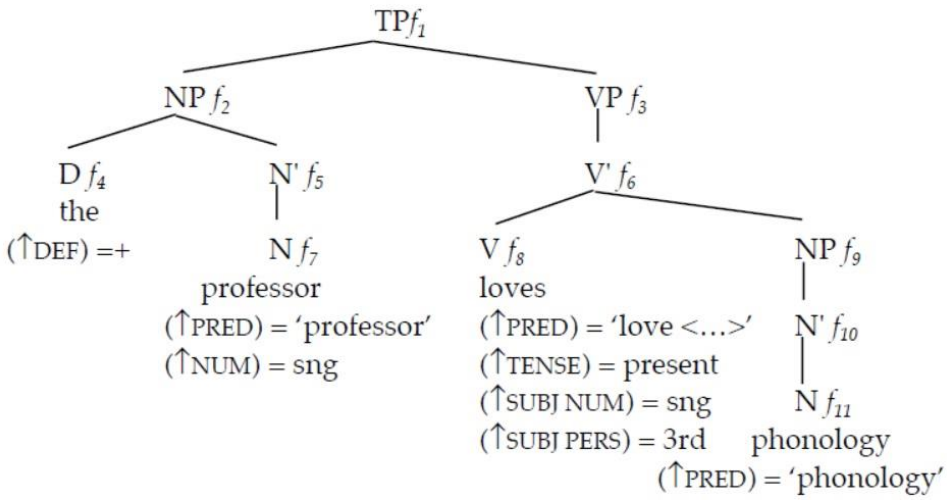


Figure 4.4 - LFG c-structure and f-structure representation[29].

Each lexical entry adds its information which are inserted in the corresponding lexical item (as for instance *professor*); while each node is characterized by a variable *f*. Correspondence is not one-to-one, thus nodes which are multiple in the tree may refer to the same AVM, i.e., f_1, f_3, f_6, f_8 (Figure 4.4).

$$
f_1, f_3, f_6, f_8, \begin{bmatrix}
\text{PRED} & \text{'love <SUBJ, OBJ>'} \\
\text{TENSE} & \text{present} \\
\text{SUBJ} & f_2, f_4, f_5, f_7, \begin{bmatrix} \text{DEF} & + \\ \text{NUM} & \text{sng} \\ \text{PRED} & \text{'professor'} \end{bmatrix} \\
\text{OBJ} & f_9, f_{10}, f_{11}, [\text{PRED} \quad \text{'phonology'}]
\end{bmatrix}
$$

Figure 4.5 - AVM for multiple nodes[29].

In Figure 5, f_2, f_4, f_5, f_7 stand for information related to SUBJ features, while f_9, f_{10}, f_{11} refer to OBJ ones. Correspondences among constituents are set up by *functional equations*, which, for example, allow the mapping between the subject of f_1 and the constituent f_2. Such functional equations, which designate the so-called *f-description,* may be used as annotations in c-structures, and, in such cases, they become *metavariables*. Metavariables,

namely variables upon variables, are of two kinds: “this node”, indicated with ↓, and “my mother” (immediately dominating node), designed with ↑. Thus, the equation ↑=↓ denotes that presenting features belong to “my mother”, namely they show a head. When such metavariables are inserted into lexical entries, they specify the same function, i.e. (↑PRED)=’love’ designates that the terminal node has the predicate value of ‘love’.

Furthermore, in LFG a principle of *unification* is effective; therefore, even if they are held into different nodes of the tree, features and functions have to be reciprocally compatible. Thus, subject features have to match with verb features, namely if *love* is 3rd singular person, then in the f-structure the subject has to be singular person. If this condition does not occur, then the f-structure will not result unified.
Three constraints are also applied to f-structure, which are represented by:

- Uniqueness, which means that a particular attribute may present at most one value.
- Completeness, which refers to the requisite that an f-structure has to hold all grammatical functions governed by its predicate.
- Coherence, namely that there is always a local predicate which governs all grammatical functions in an f-structure.

To conclude, we may say that the f-structure usage is justified by the fact that information related to a particular grammatical function “may come from more than one place in the tree and, more importantly, the sources of information do not have to be constituents” (Carnie, 2013).

3.6 Tree-Adjoining Grammar (TAG)

Tree-Adjoining Grammar (TAG) is a formalism proposed by Joshi *et al.* in 1975 and structured as a formal tree rewriting system. Several versions of such approach has been proposed, and among these the lexicalised version (LTAG), by Abeillé (1988) and Schabes (1990) and the constraint-based version (FTAG), by (Vijay-Shanker, K. 1992; Vijay-Shanker & Joshi, 1988).

TAG formalism is composed by a series of elementary trees, subdivided into initial and auxiliary ones, which state linguistic dependencies and may be

combined through sets of operations. Elementary trees "correspond to minimal linguistic structures that localize the dependencies such as agreement, subcategorization, and filler-gap" (Vijay-Shanker & Joshi 1988).

An initial tree (Figure 6) is a tree in which interior nodes are labelled with non-terminal symbols, while nodes on the frontier may be labelled with either terminal or non-terminal symbols, marked for a substitution.

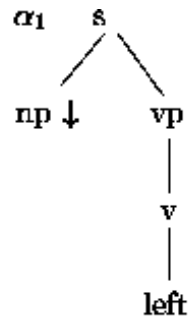


Figure 4.6 - Sample of initial tree[32].

On the other hand, auxiliary trees are outlined as initial trees, but one and only one of its border node has to be marked as foot node. Such foot node is labelled with the non-terminal symbol used for the root node (Figure 7).

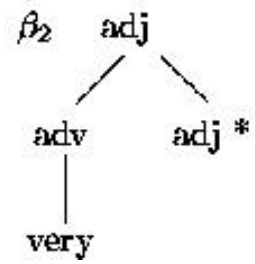


Figure 4.7 - Sample of auxiliary tree[30].

Trees may be combined through two mechanisms, namely composition operations: adjunction and substitution. Adjunction, which is not allowed on substitution nodes, inserts an auxiliary tree into a tree; on the other hand, substitution inserts a derived or elementary tree in the substitution node of a TAG tree.

[32]Image elaborated from

http://www.let.rug.nl/~vannoord/papers/diss/diss/node59.html.

In TAG approach, one tree corresponds to a rule, but not all rules are lexicalised, which means that a TAG is lexicalised if each elementary tree has at least one leaf with a terminal label.

Among linguistic principles which motive such grammar formalism, we may include:

- Predicate-Arguments Co-occurrence Principle: each predicative unit (verb, predicative noun, adjective) has in its elementary trees at least a number of substitution sites equal to the number of its arguments.
- Semantic Anchoring Principle: each elementary tree is semantically non-empty.
- Compositionality Principle: an elementary tree captures exactly one semantic unit (Gardent, 2006).

In order to provide constraints for grammar specifications and to produce only valid grammar trees, such approach uses an eXtensible MetaGrammar (XMG) formalism. XMG specifies three types of automatic (optional) mechanisms, which limits outputs trees produced by a compiler: formal, operational and language-dependent constraints.

Formal constraints ensure that the trees generated by the compiler are regular TAG trees. In addition, they state criteria which are essential for output structures, for example:

- Each node has a unique category label,
- Each leaf node is marked either as subst, as foot or as anchor ,
- The category of a foot node is identical to that of the root node (Gardent, 2006).

Based on a tree logic integrating node colours, operational constraints control the combination of tree fragments, allowing several times their multiple re-use. Thus, such constraints:

- Avoiding node naming issues: no names needed, the fusion of two nodes is controlled by colour rather than by identical global or semi-global names.

- Simplifying the grammar specification: node equations are replaced by implicit coloured node identifications.
- Reusing the same tree fragment several times.

4. Linguistic Resources

Usually, LRs may be grouped into two main classes: LRs structured as electronic dictionaries and resources developed as tagged corpora. Thus, the first class holds resources in which lexical entries are stored, independently from additional information presented in lexical databases. This means that some resources also report semantic information, i.e., sentence contexts in which lexical entries may be used, and relations with other lexical entries.

On the other hand, the second type of resources includes corpora which are tagged with semantic roles or other information

4.1 WordNet

WordNet is a lexical-semantic electronic resource of English[33], developed manually and organized in concepts and words that express such concepts. The project was initially created under the direction of George Armitage Miller at the Cognitive Science Laboratory of Princeton University. As reported in the project Web page[34], in WordNet "nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations" (Miller, 1995; Fellbaum, 1998). Thus, WordNet represents concepts as a set of (roughly) synonymous words that all refer to the same entity, event, or property (Chiarcos *et al.*, 2013b).

[33]Actually, WordNet is developed and maintained for about 80 languages. The Institute of Computational Linguistics C.N.R., in Pisa, Italy, is the developer of the Italian WordNet (ItalWordNet) (http://catalog.elra.info/product_info.php?products_id=1110), which contains about 49.500 synsets. ItalWordNet also includes a top-level ontology, formed by 63 basic semantic classes, and a domain ontology, referred to a subject-domain relationship, which is optionally assigned.

[34]Princeton University "About WordNet." WordNet. Princeton University. 2010.

http://wordnet.princeton.edu.

WordNet may seem to be similar to a thesaurus, but two of its main features help us in singularising it. The first one concerns the recording of specific senses of words, not just word forms, namely strings of letters. Such characteristic aims at disambiguating words which are close to others in the network. Indeed, each entry contains a description, provided by WordNet lexicographers, together with some samples of sentences (Figure 8) useful to disambiguate meanings.

The second feature is represented by the employment of labels to identify semantic relations among words, while thesaurus just applies a meaning similarity criterion. Synonyms are grouped inside unordered synsets, which are linked one to the other by means of conceptual relations. Furthermore, a synset also includes a definition and some samples of sentences, in which elements of the synset may be used. In WordNet, different relations among words are recognized, i.e., among these, the main is represented by synonymy.

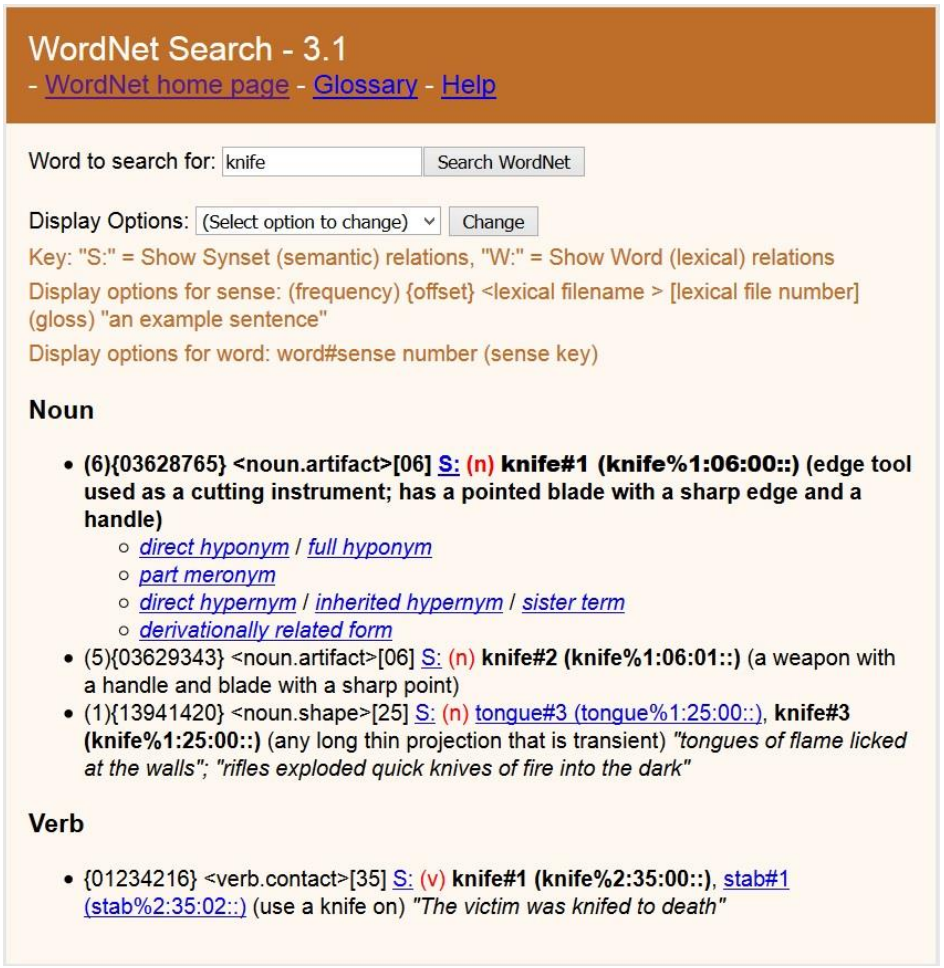


Figure 4.8 - WordNet entry for knife.

Inside WordNet and its synsets super-subordinate relations, that is hyperonymy, hyponymy or "is-a" relation[35], are the most frequent among all. These relations connect generic synsets to more specific synsets, which means that a synset containing the entry 'knife' is direct interlinked with its hypernym 'edge_tool' and with its hyponyms, such as 'bread_knife', 'butcher_knife', etc.

Thus, WordNet creates a hierarchical structure, in which all nouns refer to the root node 'entity' at last. It also classifies nouns, differentiating common

[35]In knowledge representation, "is-a" (is_a or is a) is a subsumption relationship between abstractions (e.g. types, classes), where one class A is a subclass of another class B (and so B is a superclass of A). For more information, https://en.wikipedia.org/wiki/Is-a.

nouns, also called Types, and specific elements defined Instances, such as persons, countries and so on, which are terminal nodes in their hierarchies.

As Figure 8 shows, another kind of recognized relation is meronymy, namely the part-all relationship, which connects 'blade' or 'haft' to 'knife'.
Verb synsets are also organized in hierarchies, and such classification allows to discriminate increasing specification levels of an event, e.g., 'separate'-'cut'-'hack'.

Adjectives are systematized applying an antonymy mechanism, which means that they are defined as one of the elements in a pair of words, presenting an opposite meaning. Pair of words may be direct or indirect antonyms. Direct antonyms are related by a strong semantic component, e.g. 'sweet'-'sour'. On the other hand, indirect antonyms just present a semantic similarity, e.g. 'sweet-'cloying'.

WordNet library presents a unidirectional morphological component, which means that WordNet does not offer plural or inflected forms. The morphological component presents some simple inflectional rules, which are applied until it obtains a correspondence with a word form contained in WordNet. Basically, such component evaluates inputs as valid inflected forms, thus it accepts forms that are not a word. This means that, according to general inflectional rules, it derives 'childes' from 'child' (obviously, a set of irregular forms, which contains 'children', exists).

Furthermore, only content words (i.e., noun, verbs, adjectives, and adverbs) are included in WordNet, which means that function words (as for instance determiners, prepositions, pronouns, conjunctions, and particles) are omitted (Mille, 1995; Fellbaum, 1998).

Several tools are available to compute the semantic distance between two synsets, as for instance the open source module WordNet::Similarity[36], which offers different similarity and semantic relatedness measures, based on WordNet.

In their work, Budanitsky & Hirst (2006) propose an evaluation of lexical semantic relatedness based on WordNet measures. Such evaluation starts from the assumption, derived from Morris & Hirst (1991), that five types of semantic relations may exist between two words. Indeed, two words are considered related or semantically close, when any of the following conditions is satisfied:

[36]http://wn-similarity.sourceforge.net/.

- Two words have a category in common in their index entry
- One word has a category in its index entry that contains a pointer to a category of the other word.
- One word is either a label in the other index entry or is in a category of the other.
- Both two words contained in the same category
- Both two words have categories in their index entries that point to a common category.

4.2 FrameNet

FrameNet is an ongoing project of Berkeley University, risen by Charles Fillmore, the main theorist of Frame Semantics. Fillmore's idea rests on the principle that a semantic theory has to be founded completely on human comprehension processes, namely on the way in which we understand discourses in contexts. In order to achieve this purpose, it is necessary to combine different information, such as word meanings, grammatical properties and real-world knowledge (Goddard, 2011).

Goddard declares that "according to frame semantics, the meaning of a word can only be understood against a background frame of experience, beliefs, or practices that 'motivate the concept that the word encodes'".

FrameNet represents an on-line lexical resource for English[37], carried by corpus evidence. "The aim is to document the range of semantic and syntactic combinatory possibilities - valences[38] – of each word in each of its senses, through computer-assisted annotation of example sentences, together with automatic tabulation and display of the annotation results" (Ruppenhofer *et al.,* 2006).

The project has also produced a lexical database, which includes more than 10,000 lexical units, more than 6,000 of which are fully annotated, in

[37]Recently, the project has been extended to several other languages, e.g. Spanish, German and Japanese.

[38]The term valence stands for semantic and syntactic combinatory possibilities.

nearly 800 hierarchically related semantic frames, exemplified in more than 135,000 annotated sentence[39].

In FrameNet, each lexical unit is composed by a matching between a word and a meaning. Such meaning is inserted in a semantic frame, which represents a conceptual structure, able to depict a certain kind of situation, object or event with its participants and props.

Thus, semantic frames are developed using a set of Frame Elements and a Frame Definition that specifies how Frame Elements are interconnected. Frame elements may be situation-specific semantic or generic roles, such as Agent, Patient and Instrument. Furthermore, there are extra-thematic frame elements, as for instance Manner, Time, Reason, Duration, Circumstances and Reciprocation[40].

Simple cases are those in which frames evoking a lexical unit are verbs, and frame elements are their syntactic dependents, such as in the following sample:

[$_{\text{Cook}}$ Matilde] **fried** [$_{\text{Food}}$ the catfish] [$_{\text{Heating\_instrument}}$ in a heavy iron skillet][41].

Such annotations allow to derive lexical entries for predicating words, from which frames, underlying meanings, are identified. They also identify structures, directed by words, which realize frame elements. “The main purpose of annotating such items is to identify the most common predicates that govern phrases headed by them, and thus to illustrate the ways in which these common nouns function as FEs within frames evoked by the governing predicates” (Goddard, 2011:5).

In FrameNet, sentences are annotated with triple constellations which describe frame element realizations through:

- A frame element (as for instance Food),
- A grammatical function (as for instance Object),
- A phrase type (as for instance NP).

[39]Such numbers refer to 2010.

[40]Such task is generally well-known as semantic role labelling, or shallow semantic parsing.

[41]Sample taken from Goddard (2011).

FrameNet annotations are obtained using two methodologies. Given a target lexical unit, the first way consists in extracting sentences from several texts of a corpus that contain such lexical unit. Subsequently, selections of the extracted sentences are annotated according to the target lexical unit. This process aims at recording valence ranges of each word in each of its senses. On the other hand, the second methodology is based on annotating running text. These two kinds of annotation differ mainly in the way sentences are chosen: indeed, for running text they are chosen by the author of the text. In the annotation layering technique, used for running text, "FN lexicographers can one by one declare each word in a sentence a target**,** select a frame relative to which the new target is to be annotated, get a new set of annotation layers (frame element, grammatical function, phrase type) and appropriate frame element tags, and then annotate the relevant constituents" (Ruppenhofer *et al.,* 2006).

As reported in Marano (2012:93), FrameNet is structured on specific characteristics, which differentiate it from other LRs, as for instance WordNet:

- "lexical units are provided with definitions taken from Oxford paper dictionary entries;
- multiple annotated sample sentences are given for each lexical unit and its senses;
- sample sentences are taken from concrete corpora and are not arbitrarily constructed;
- English lexicon analysis is achieved frame by frame rather than lemma after lemma; this helps in avoiding the use of the traditional alphabetic description/completion, which does not always support the correct explanation of word combinatorial and semantic characteristics;
- each lexical unit is not also linked to a given semantic frame, but also to all the other semantically similar words by which that frame is brought to mind;
- while WordNet and all ontologies are based on hierarchical relations between nodes, FrameNet uses a network of relations between frames,

the most important of which are 'inheritance', 'using', 'subframe', and 'perspective on'"[42].

4.3 VerbNet

As stated in its Web site[43], "VerbNet (VN) (Schuler, 2005) is the largest on-line verb lexicon currently available for English. It is a hierarchical domain-independent, broad-coverage verb lexicon with mappings to other lexical resources such as WordNet (Miller, 1995; Fellbaum, 1998), Xtag (XTAG Research Group, 2001), and FrameNet (Baker *et al*., 1998). VerbNet is organized into verb classes extending Levin (1993) classes through refinement and addition of subclasses, to achieve syntactic and semantic coherence among members of a class".

In VerbNet, thematic roles, selection restrictions on the arguments and frames are provided for each verb class. Indeed, a verb class includes a set of syntactic descriptions, also called syntactic frames, which describe the possible argument structure for sentence construction, i.e., transitive, intransitive, prepositional phrase, and so on. The types of thematic roles, allowed by arguments, are also dependent on semantic restrictions. This means that a semantic restriction (such as animal, human, inanimate, etc.) is used as constrain in order to define constituents and also their syntactic structures, which may be associated to a thematic role.

As acknowledged in Kipper *et al.* (2006), the original classification was proposed by Levin (1993).

[42]For more on these relations, see Ruppenhofer et al. (2006)

[43]http://verbs.colorado.edu/~mpalmer/projects/verbnet.html.

Class Hit-18.1			
Roles and Restrictions: Agent[+int_control] Patient[+concrete] Instrument[+concrete]			
Members: bang, bash, hit, kick, ...			
Frames:			
Name	Example	Syntax	Semantics
Basic Transitive	Paula hit the ball	Agent V Patient	cause(Agent, E)manner(during(E), directedmotion, Agent) !contact(during(E), Agent, Patient) manner(end(E),forceful, Agent) contact(end(E), Agent, Patient)

Figure 4.9 - Simplified VerbNet entry for Hit-18.1 class[44].

"VerbNet (Kipper, Dang, and Palmer, 2000; Kipper, Palmer, and Rambow, 2002) extends Levin's classes by adding an abstract representation of the syntactic frames for each class, with explicit correspondences between syntactic positions and the semantic roles they express, as *in Agent REL Patient*, or *Patient REL into pieces* for *break*". (Palmer et al., 2005) As stated in Kipper *et al.* (2000), we may consider these aspects as notational variants of elementary trees, or partial derivations in TAG approach.

Thus, starting from Levin's classification, VerbNet establishes thematic labels, syntactic frames and class descriptions with their semantic predicates.

4.4 Penn TreeBank

As an eight-year project (1989-1996), the Penn TreeBank[45] is a parsed corpus, syntactically and semantically annotated, which produces:

- 7 million words of part-of-speech tagged text,
- 3 million words of skeletally parsed text,
- Over 2 million words of text parsed for predicate-argument structure,
- And 1.6 million words of transcribed spoken text annotated for speech disfluencies (Taylor *et al.*, 2003).

Its corpus is composed of texts, derived from different sources, as for instance Wall Street Journal articles, IBM computer manuals, and also

[44]Image taken from http://verbs.colorado.edu/~mpalmer/projects/verbnet.html.

[45]The name Penn TreeBank derives from the concept of TreeBank, a term coined by linguist Geoffrey Leech in the 1980s, by analogy to other repositories such as a seedbank or blood bank. Such TreeBank has been developed by University of Pennsylvania (Penn).

transcribed telephone conversations. All data produced by the Treebank is released through the Linguistic Data Consortium (LDC)[46].

Word annotation process is achieved through a two-step procedure, which involves an automatic tagging and a human correction.
As specified in the Penn TreeBank guideline:

> *Our approach to developing the syntactic tagset was highly pragmatic and strongly influenced by the need to create a large body of annotate material, given limited human resources. The original design of the Treebank called for a level of syntactic analysis comparable to the skeletal analysis used by the Lancaster Treebank… no force distinction between arguments and adjuncts. A skeletal syntactic context-free representation (parsing)* (Taylor *et al.,* 2003:23).

Therefore, the Penn TreeBank tagsets are as large as articulated and they are based on that of the Brown Corpus[47], even if there are some important differences. The motivation under the development of such structured tagsets is traceable in the work of Garside (1988), as declared in (Taylor *et al.*, 2003). Their approach aims at "the ideal of providing distinct codings for all classes of words having distinct grammatical behaviour". One difference concerns the reduction of lexical and syntactic redundancies, i.e., in the Brown Corpus tagset some of the POS tags refer uniquely to one lexical item. Thus, the Penn Treebank reduce such lexical redundancy. Furthermore, "distinctions recoverable with reference to syntactic structure were also eliminated. For instance, the Penn Treebank tagset does not distinguish subject pronouns from object pronouns, even in cases where the distinction is not recoverable from the pronoun's form, as with *you*, since the distinction is recoverable on the basis of the pronoun's position in the parse tree in the parsed version of the corpus" (Taylor *et al.*, 2003:6).

[46]http://www.ldc.upenn.edu/.

[47]The Brown University Standard Corpus of Present-Day American English (or just Brown Corpus) was compiled in the 1960s by Henry Kucera and W. Nelson Francis at Brown University, Providence, Rhode Island as a general corpus (text collection) in the field of corpus linguistics. It contains 500 samples of English-language text, totalling roughly one million words, compiled from works published in the United States in 1961. Source: https://en.wikipedia.org/wiki/Brown_Corpus.

Another difference between such two tagsets, and in our opinion one of the most significant, is related to the usage of syntactic contexts. Indeed, the Brown Corpus words are tagged independently from their syntactic functions, which means that a word is not considered referring to the phrase in which it may occur.

On the other hand, the Penn Treebank tries to encode word syntactic functions in their POS tags, in order to use the corpus as the basis for a bracketed version of the corpus itself.

A further difference concerns the usage of multiple tagging, if annotators are unsure about the correct tag, or a POS cannot be assigned to a specific word. In this way, the Penn TreeBank manages the indeterminacy which may affect POS handling, that is to say the issue of POS ambiguity, when it is not resolvable with reference to the linguistic context.

The Penn TreeBank tagset contains 36 POS tags plus 12 more tags indicating punctuation and currency symbols (Figure 4.10). As previously mentioned, such tags are applied through an automatic assignment and a manual correction.

CC	Coordinating conj.	TO	infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present pple
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sg. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sg. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	"	Straight double quote
RBR	Adverb, comparative	`	Left open single quote
RBS	Adverb, superlative	``	Left open double quote
RP	Particle	'	Right close single quote
SYM	Symbol	''	Right close double quote

Figure 4.10 - The Penn TreeBank POS tagset[48].

During the early times of the project, the automatic tagging step was led by a stochastic algorithm, developed at AT&T Bell Labs, and called PARTS (Church, 1988). "The output of PARTS was automatically tokenized and the tags assigned by PARTS were automatically mapped onto the Penn Treebank tagset" (Marcus *et al.*, 1993). In recent times, the automatic POS assignment is achieved through a cascade of stochastic and rule-driven taggers.

The second step is performed by annotators which correct errors of POS automatic tagging.

During the project, a test to maximize speed, inter-annotator consistency, and accuracy has been performed. The result of such test is reported by the authors, and may be resumed as follows: "This experiment showed that manual tagging took about twice as long as correcting, with about twice the inter-annotator disagreement rate and an error rate that was about 50% higher" (Marcus *et al.*, 1993:319).

[48]Table taken from Taylor *et al.* (2003).

In order to accomplish the tagging task, in the Penn TreeBank, two kinds of syntactic bracketing have been applied: a skeletal context-free bracketing and a predicate-argument structure. The first one, employed during the early phase of the project, presents limited empty categories and no information about non-contiguous structures and dependencies.

Subsequently, due to the need of a different level of representation, TreeBank II has been introduced for providing a form of predicate-argument structure. Such new kind of annotation introduces further information:

- "A clear, concise distinction between verb arguments and adjuncts where such distinctions are clear, with an easy-to-use notational device to indicate where such a distinction is somewhat murky.
- A non-context free annotational mechanism to allow the structure of discontinuous constituents to be easily recovered.
- A set of null elements in what can be thought of as "underlying" position for phenomena such as wh-movement, passive, and the subjects of infinitival constructions, co-indexed with the appropriate lexical material" (Taylor *et al.*, 2003:9).

The predicate-argument scheme has been introduced in order to provide a correct semantic label to each argument of the predicate (Figure 4.11). Such semantic label allows to categorize argument role with respect to that predicate (subject, object, etc.), differentiating the arguments of the predicate, and adjuncts of the predication.

Text Categories	
-HLN	headlines and datelines
-LST	list markers
-TTL	titles
Grammatical Functions	
-CLF	true clefts
-NOM	non NPs that function as NPs
-ADV	clausal and NP adverbials
-LGS	logical subjects in passives
-PRD	non VP predicates
-SBJ	surface subject
-TPC	topicalized and fronted constituents
-CLR	closely related - see text
Semantic Roles	
-VOC	vocatives
-DIR	direction & trajectory
-LOC	location
-MNR	manner
-PRP	purpose and reason
-TMP	temporal phrases

Figure 4.11 - Functional Tags[49].

By means of this two-step procedure, consisting of automatic annotation and manual correction, the Penn TreeBank produces three types of labelling: POS tagging, syntactic bracketing, and disfluency annotation[50].

[49]Table taken from Taylor *et al.* (2003).

[50]This last type of annotation refers to the final project undertaken by the Treebank (1995-6), which we do not deal with in this dissertation. Such project aims at creating "tagged and parsed version of the Switchboard corpus of transcribed telephone conversations, along with a version which annotated disfluencies which are common in speech (fragments of words, interruptions, incomplete sentences, fillers and discourse markers).

The disfluency annotation system (based on Shriberg (1994)) distinguishes complete utterances from incomplete ones, labels a range of non-sentence elements such as fillers, and annotates restarts" Taylor *et al.* (2003) p.15.

```
Battle-tested/JJ Japanese/JJ industrial/JJ managers/NNS
here/RB always/RB buck/VBP up/RP nervous/JJ newcomers/NNS
with/IN the/DT tale/NN of/IN the/DT first/JJ of/IN
their/PP$ countrymen/NNS to/TO  visit/VB Mexico/NNP ,/,
a/DT boatload/NN  of/IN samurai/FW warriors/NNS  blown/VBN
ashore/RB 375/CD years/NNS ago/RB ./.
```

Figure 4.12 - Sample of POS tagging result[51].

Even if the Penn TreeBank project is no longer in operation, it has produced a large amount of data which represent even now a significant LR, largely employed in various NLP researches and tasks.

4.5 PropBank

PropBank[52], which stands for Proposition Bank, is the name of a project in Automatic Content Extraction (ACE)[53], which aims at creating a corpus of text annotated with information about basic semantic propositions. Basically, PorpBank adds predicate-argument relations to the syntactic trees of the Penn Treebank, presenting them in a single instance which contains information about the location of each verb, plus the location and the identity of its arguments.

Such information have been organized in the form of variables, which represent:

- Location information
- Annotator information
- Inflection information
- Roleset identifier
- Verb (that is predicate) location
- Argument location and types

[51]Reworked version of image taken from Taylor et al. (2003).

[52]http://verbs.colorado.edu/~mpalmer/projects/ace.html.

[53]ACE is a research program, started with a pilot study in 1999, for developing advanced Information extraction technologies.

https://en.wikipedia.org/wiki/Automatic_Content_Extraction.

PropBank annotates predicates applying the concept of semantic roles[54]. The motivation may be found in Palmer *et al.* (2005) "While the TreeBank provides semantic function tags such as temporal and locative for certain constituents (generally syntactic adjuncts), it does not distinguish the different roles played by a verb's grammatical subject or object in the above examples".

Semantic role annotation is achieved through a semi-automatic process; indeed, it is not possible to use a fully automatic process because it does not guarantee a 100% accuracy (Palmer et al., 2005). Thus, the annotation process starts with a rule-based automatic tagging, the results of which are subsequently hand-corrected. Such an annotation process is based on a classification, suitable to describe sentence contexts which constitute the frame in which words may occur. Actually, as we have seen in 4.3, Levin (1993) proposes a verb classification defining classes on the basis of verb capability to occur or not to occur in specific frames. According to Levin, frames are mainly syntactic; however, they also contain a semantic component which is reflected into the constraints of any allowable arguments.

The main aim of PropBank is to provide a representation of syntactic alternation and an annotated corpus of data which supports empirical study. Even if Levin's classification and VerbNet provides information about alternation patterns and their semantics, they do not offer a quantification of frequency of such alternations and either on their effect on language understanding systems (Palmer *et al.,* 2005).

It is worth stressing that a universal set of semantic roles, which is able to include all predicates, is a rather challenging mission to achieve. For this reason, in PropBank, definitions of semantic roles are delineated through a verb-by-verb analysis. Thus, allowable arguments for each verb are identified and numbered, starting from a 0 value. For example, *Arg0* stands for the argument exhibiting features of a prototypical Agent (Dowty, 1991). A sample of verb-specific numbered role for the entry *accept* is provided in the following *Frameset*:

[54]Semantic role labeling, sometimes also called shallow semantic parsing, is a task in natural language processing consisting of the detection of the semantic arguments associated with the predicate or verb of a sentence and their classification into their specific roles.

Source: https://en.wikipedia.org/wiki/Semantic_role_labeling.

Frameset **accept.01** "take willingly"
Arg0: Acceptor
Arg1: Thing accepted
Arg2: Accepted-from
Arg3: Attribute
Ex: [Arg0 He] [ArgM-MOD would] [ArgM-NEG n't] *accept* [Arg1 anything of value] [Arg2 from those he was writing about]. (wsj0186)[55]

A *Frameset* corresponds to a roleset and its associated frames. We may describe a roleset as the representation of a distinctive usage of a verb entry. A roleset may be also associated with frames which describe the syntactic variations that are applicable to that set of roles. Obviously, if a verb is polysemic, it may present different Framesets, which describe the different sets of roles, required for the different meanings of the verb. In other words, different meanings are differentiated in more Framesets, on the basis of semantic and syntactic criteria and of the number(s) of arguments, required by each meaning.

PropBank annotations also provide samples (Ex) of verb usages, in order to cover the range of syntactic alternations of a specific roleset. Thus, verb entries include all information about the semantic roles described in the role sets. In addition, verbs may also present a set of general arguments, defined as adjunct-like arguments (ArgMs), divided into the following sub-categories:

- Location (LOC),
- Cause (CAU),
- Extent (EXT),
- Time (TMP),
- Discourse connectives (DIS),
- Purpose (PNC),
- General-purpose (ADV),
- Manner (MNR),
- Negation marker (NEG),
- Direction (DIR),
- Modal verb (MOD).

[55]Example taken from Palmer *et al.* (2005).

EXT and PRD tags indicate numbered arguments. Indeed, EXT stands for a constituent which is a numerical argument on its verb, e.g. 'I would walk *500 miles*', while PRD indicates a secondary predication, that is it describes a more fine-grained relation.

As stated in Palmer *et al.* (2005), it is worth stressing that "although they are not considered adjuncts, NEG for verb-level negation (e.g., 'John did*n't* eat his peas') and MOD for modal verbs (e.g., 'John *would* eat everything else') are also included in this list, to allow every constituent surrounding the verb to be annotated. DIS is also not an adjunct, but was included to ease future discourse connective annotation" (Palmer *et al.,* 2005:6).

Semantic roles are assigned by annotators labelling nodes with roles in the syntactic trees of the Penn TreeBank. Even if the syntactic parse may not be changed, annotators may apply labels without other limits.

4.6 Linked Open Data (LOD) and Linguistic Linked Open Data (LLOD)

The term Linked Open Data (LOD) refers to the publication of dataset of structured information, mainly in RDF, and in a way that they may be reciprocally interlinked. In other words, LOD aims at creating a connected network of data, which are structured and related, in order to improve information processing by means of machines.

A subset of LOD is represented by Linguistic LOD (LLOD), which amounts to Linguistic and Open resources in RDF format interlinked with other Linguistic and Open resources. Thus, LLOD are open LRs, namely LRs to which the LOD paradigm has been applied, even if they are not too many as for LOD**.**

Therefore, LOD and LLOD goals may be summed as the attempt to create interoperability and information integration among data, which means the possibility of retrieving and combining information derived from different sources.

Then, interoperability and information integration of any kind of data on the Web may be defined as the main goal of publishing open data[56]. For such

[56]The concept of Open Data is not new, but its formalized definition is latest. We propose the Open Definition:

reason, Bizer *et al.* (2009) propose some best practice rules, namely some linked data principles, that we should follow to achieve data interoperability and information integration:

1. Use Uniform Resource Identifiers (URIs) as (unique) names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using Web standards such as RDF and SPARQL.
4. Include links to other URIs, so that they can discover more things.

Such linked data principles have been applied also for the handling of lexical and linguistic data. Indeed, this is recognized as an efficient approach to guarantee interoperability and information integration also for LRs on the Web.

As for the previous four-step list, the first principle entails the need of assigning URIs to identify each resource element. Chiarcos *et al.* (2013b) define such procedure as an advantage that makes resources "uniquely and globally identifiable in an unambiguous fashion".

The second principle refers to the necessity of retrieving information about resources using a human-readable and browseable view, such HTML allows to do.

The third rule requires the employment of standards, both to represent resources (as for instance Resource Description Framework - RDF) and to query online repositories (as for instance SPARQL Protocol and RDF Query Language)[57].

The last principle aims at creating a network among resources; indeed, adding links to other URIs represents the way in which we may create a cloud of online resources (as for instance Figure 13).

Following these principles, LOD are structured as repositories of data encoded in RDF triples. Indeed, in linked data definition, RDF has a central role, since it allows the use of a data model which is based on labelled directed (multi-)graphs. Such data model has been created for providing metadata

"Open means **anyone** can **freely access, use, modify, and share** for **any purpose** (subject, at most, to requirements that preserve provenance and openness)." Thus, "Open data and content can be **freely used, modified, and shared** by **anyone** for **any purpose**". For more information, visit: http://opendefinition.org/.

[57]We will deal with RDF and SPARQL extensively in Chapter VI.

about resources both offline and online. RDF represents information about resources using triples which are formed by:

- A subject which is a resource, in graph-theoretical terms a labelled node
- A property which represents a relation, in graph-theoretical terms a labelled edge, associating a subject and its object
- An object which stands for another resource, or a literal, as for instance a string.

Nodes are identified by URIs, so that every resource and property may be recognized uniquely, since they become globally unambiguous in the web of data.

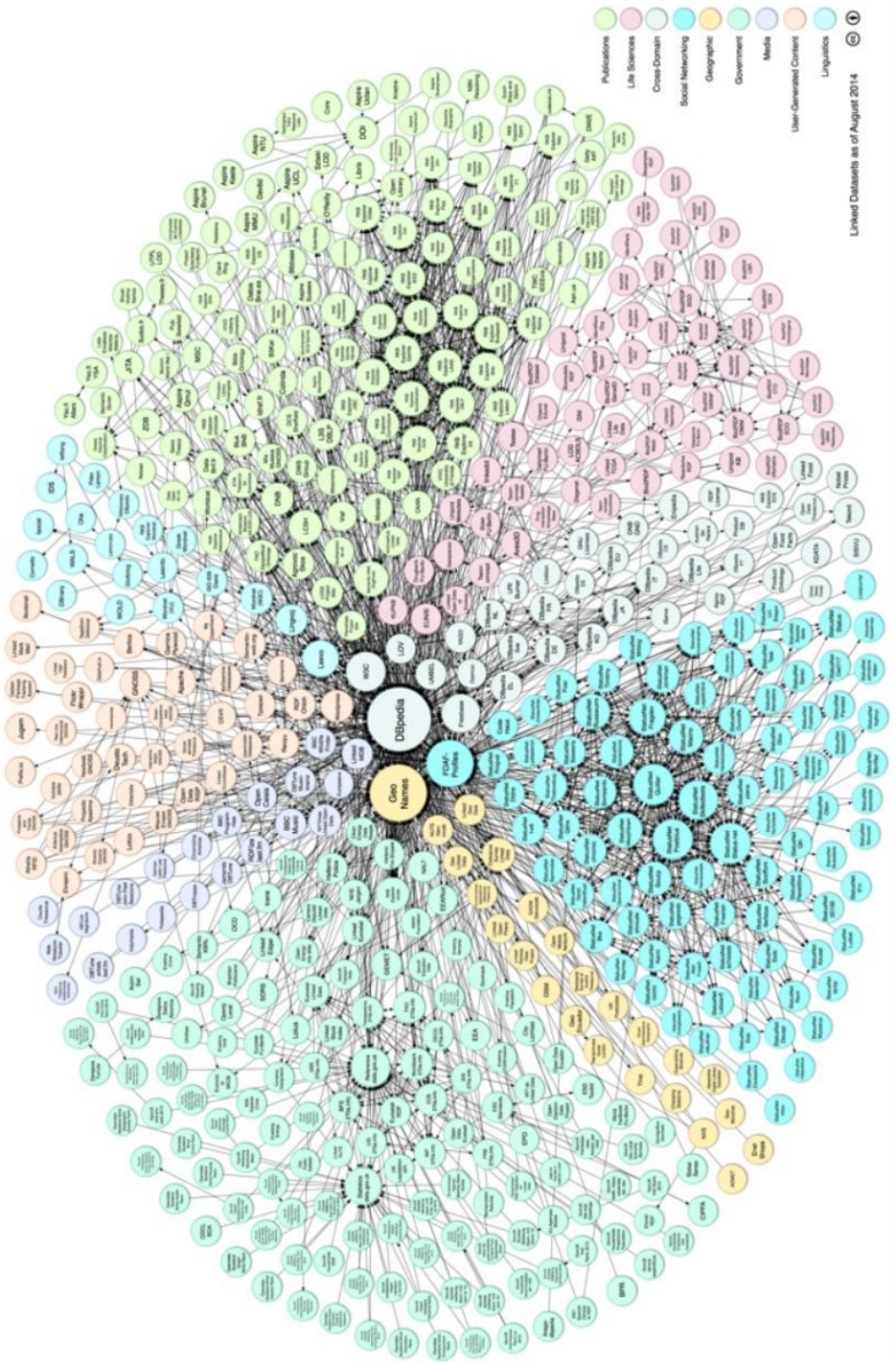


Figure 4.13 – Linked Dataset as of August 2014[58].

[58]Image taken from http://lod-cloud.net/.

Actually, in the last years, is arisen an important challenge about storing, connecting and exploiting the wealth of language data assembled in half a century of computational linguistics research (Chiarcos *et al.*, 2013b). The first aim, the interoperability of language resources, seems to have been partially solved (Ide & Pustejovsky, 2010) by means of LLOD project.

The LLOD cloud (Figure 14) is a collaborative project developed by numerous members of the Open Linguistic Working Group OWLG[59] (Chiarcos *et al.*, 2012) who aims at creating a LOD cloud of LRs. Indeed, the main reason for this project is that for linguistics much data are published using proprietary and not open formats. Chiarcos, *et al.* (2013b) claim that "modelling and publishing language resources as linked data offers crucial advantages as compared to existing formalisms".

A certain number of LRs, mainly vocabularies, has been developed using RDF, and a part of these may be directly applied to LRs. Such vocabularies should hold labels, which may represent data, and additional constraints which introduced in order to formalize specialized RDF sub-languages. This is the case with the Web Ontology Language (OWL) which specifies which types of data are required for representing ontologies as an RDF extension, i.e., classes (concepts), instances (individuals) and properties (relations) (Chiarcos *et al.*, 2013).

As also reported in the project web site[60], the primary benefits of LLOD have been identified as:

1. Representation: Linked graphs are a more flexible representation format for linguistic data

[59]OWLG, founded in 2010, is an open network of individuals interested in linguistic resources. The group goals are mainly:

- Promote open data in relation to language data
- Facilitate communication among researchers which use, distribute or maintain open linguistic data
- Mediate between providers and users of technical infrastructures.

https://www.w3.org/community/ld4lt/wiki/images/3/37/Lider-workshop-20140509-olwg-chiarcos.pdf.

http://linguistics.okfn.org/2011/05/20/the-open-linguistics-working-group/.

[60]http://www.linguistic-lod.org/.

2. Interoperability: Common RDF models can easily be integrated
3. Federation: Data from multiple sources can trivially be combined
4. Ecosystem: Tools for RDF and linked data are widely available under open source licenses
5. Expressivity: Existing vocabularies such as OWL, Lemon and NIF help express LRs.
6. Semantics: Common links express what you mean.
7. Dynamicity: Web data can be continuously improved.

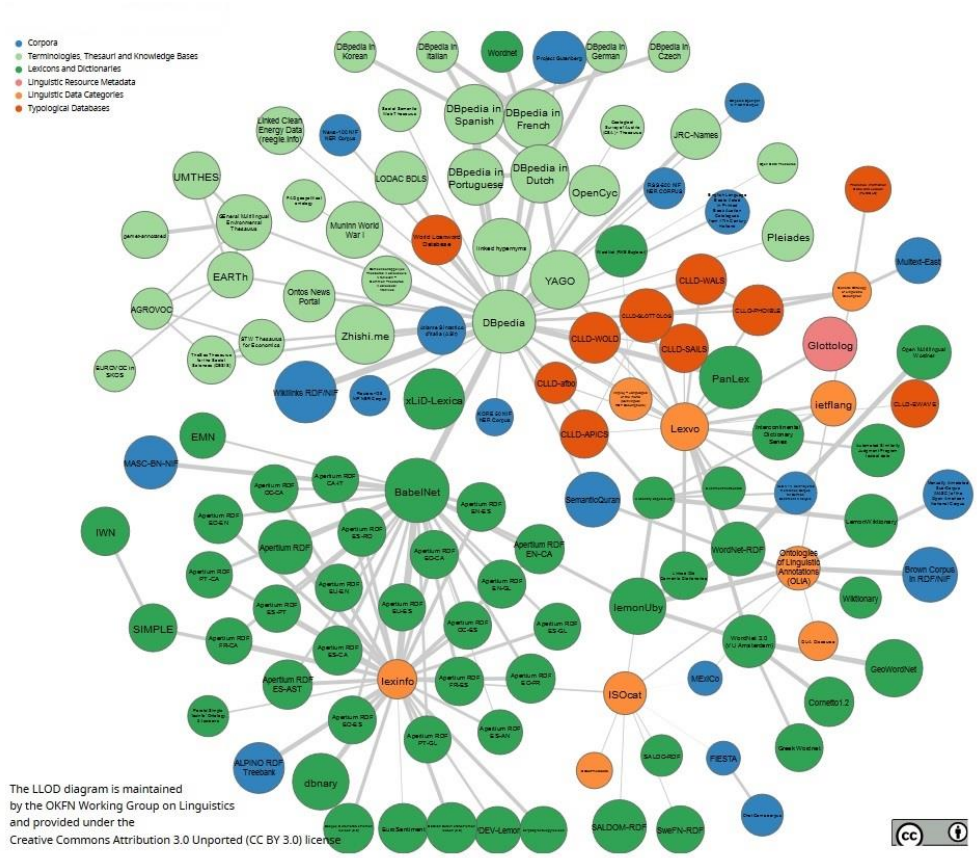


Figure 4.14 - Linguistic Linked Open Data Cloud[61].

Modelling LRs as LLOD is a two-step process in which we firstly specify data structures and consistence constraints over these, and subsequently we

[61]"Open Linguistics Working Group (2012), The Linguistic Linked Open Data cloud diagram (draft).

Image taken from http://linguistic-lod.org/llod-cloud#.

convert data into such representations. As we have seen in the previous pages, LRs may be distinguished into two classes: lexical-semantic resources and annotated corpora. The first class refers to those resources which include lexemes and information about them and about relations with other lexemes.

On the other hand, annotated corpora are represented by textual data annotated with their linguistic characteristics.

Both kinds of LRs may be represented and described through RDF labelled directed (multi-)graphs. As stated in Chiarcos *et al.* (2013b:11) "Unlike other graph-based modelling formalisms applied to language resources, e.g., GraphML[62] (Brandes *et al.*, 2010), RDF provides additional means to formalize specific data types, and thereby to establish a reserved vocabulary and to introduce structural constraints for nodes, edges or labels".

Modelling LRs in RDF provides several advantages, such as the possibility of creating linkages among such resources in order to improve the quality of querying across resources. In other words, RDF model allows to represent different LRs in a uniform way, guaranteeing the interoperability among data, which means that we are able to retrieve information from different linguistic sources and repositories.

It is important to stress that RDF model usage has been growing up in recent years, mainly in Cross-Lingual Information Retrieval (CLIR) and Machine Translation (MT). Indeed, several researches are focused on the opportunity offered by the creation of a shared data model which may advance both IR and translation. Using RDF representation in IR tasks promises an improvement in

[62]GraphML is based on XML and it is "a comprehensive and easy-to-use file format for graphs. It consists of a language core used to describe the structural properties of a graph, and of a flexible extension mechanism used to add application-specific data. Its main features include support of

- directed, undirected, and mixed graphs,
- hypergraphs,
- hierarchical graphs,
- graphical representations,
- references to external data,
- application-specific attribute data, and
- light-weight parsers".

Source: http://graphml.graphdrawing.org/.

terms of precision and recall, due to the fact that queries can be managed against multi-lingual repositories and KBs. At the same time, MT may take advantage from cross-lingual mappings, as argued in different works (Gracia *et al.*, 2012, 2012b; Buitelaar *et al.*, 2012; Monti *et al.*, 2013; di Buono *et al.*, 2013b).

In Chapter V, we will show how we combine RDF model with the development of our LRs in a system workflow able to manage query representations using SPARQL Protocol and RDF Query Language expressions.

In addition, together with the spread of LOD and LLOD, we note the development of sophisticated NLP applications, capable to integrate such synergic tools. Indeed, as also stated in Hellmann *et al.* (2013), "many NLP tasks can greatly benefit from making use of this wealth of knowledge being available on the Web in structured form as Linked Open Data (LOD). The precision and recall of Named Entity Recognition, for example, can be boosted when using background knowledge from DBpedia[63], Geonames[64] or other LOD sources as crowdsourced, community-reviewed and timely-updated gazetteers".

Obviously, the integration of these resources and tools requires time and strong efforts, together with the overcoming of serious challenges such as semantic alignment, identification and provenance (see Chapter V).

As we have argued in Chapter III, the use of linguistic rule-based routines in NLP does not concern such a wide range of research-field applications, while it is most used in industrial ones.

A few examples of rule-based approaches to NLP components or systems include:

- Any tokenizer with morphological parsing rules, regular expressions, or character classes.
- A set of search terms used to categorize, tag, or label documents with metadata.

[63]For more information, see: www.dbpedia.org.

[64]Geonames is geographical database covers all countries and contains over eight million place names. http://www.geonames.org/.

- Information Extraction rules or Context Free Grammar rules, including those using Boolean operators, context operators, or regular expressions.
- Any lexicon, word-list, or KB.

In the chapter which follows, we will examine how a rule-based approach and human effort in language formalizations seem to guarantee an improvement in all the NLP tasks.

V – NLP FOR ONTOLOGY LEARNING AND POPULATION IN THE CULTURAL HERITAGE DOMAIN

Some people, when confronted with a problem, think "I know, I'll use regular expressions." Now they have two problems.
Jamie Zawinski

In the previous chapters, we have introduced the main models and methods used in NLP tasks and, specifically, in OL and population. As for these topics, in this chapter, we will present our approach to the achievement of natural language formalizations and our proposal for the development of efficient and effective KR and KE.

On such premises, we present an approach, based on Lexicon-Grammar (LG) framework, which aims at improving KR and KE in the Archaeological domain. We intend to demonstrate how our language formalization technique can be applied to both process and populate a domain ontology.

1. Lexicon-Grammar for KR and KE

As for KR and KE, we present here an approach based on Lexicon-Grammar (LG) framework, in this way aiming to improve these two tasks in the Archaeological domain. LG main goal is to describe all mechanisms of word combinations closely related to concrete lexical units and sentence creation, and to give an exhaustive description of lexical and syntactic structures of natural language. The study of simple or nuclear sentences is achieved analysing the rules of co-occurrence and selection restriction, i.e. distributional and transformational rules based on predicate syntactic-semantic properties[1].

We intend to demonstrate how LG language formalization technique can be applied to KR and KE processes in order to populate a domain ontology.

[1]As we will see, LG co-occurrence and selection-restriction rules may be also described by means of RDF graphs.

Indeed, as we have seen in the previous paragraphs, in the last years many approaches to KR and KE tasks have been developed, some of these being concept-based, which means that they employ a reduced number of features in order to represent and extract semantic content. Other approaches are focused on the development of a representation of terms inside a semantic space, suitable to infer the meanings and behaviours of a given word/phrase.

More recent techniques include domain ontology-based approaches, due to the fact that "ontologies reflect the structure of the domain and constrain the potential interpretations of terms" (Sánchez Cisneros & Aparicio Gali, 2015).

Therefore, the use of ontologies in the processes of semantic representation and extraction seems a promising and challenging field, mainly when we want to process the contents of different KBs or unstructured texts.

Starting from the assumption that a coherent and consistent language formal description is crucial and indispensable to achieve a correct semantic representation of whatsoever knowledge domain, our research focuses on a hybrid approach to content analysis and IE. Such an approach is based on language formal description and takes into account the fact that terminological and specialized Atomic Linguistic Units (ALUs) may also be interlinked with, or refer to, other knowledge domains.

In fact, our idea also springs from Bachimont (2000) who states that "defining an ontology for knowledge representation tasks means defining, for a given domain and a given problem, the functional and relational signature of a formal language and its associated semantics". Actually, we will see that extracting information from unstructured texts brings critical challenges for the application of ontology population and knowledge representation techniques. Therefore, in this perspective, terminology mining becomes relevant in order to both manage domain knowledge and guarantee the maintenance and updates of an ontology.

Furthermore, some knowledge domains, as the Archaeological one, present a range of variable types and properties of contents, due to the fact that they are characterized by a strong Semantic Expansion (SE), being strictly interlinked with other domains.

Our linguistic formalization is based on an accurate observation of these properties, and on an appropriate linguistic data recording of all lexicon and lexical entry combinatory behaviours, encompassing syntax and, also, lexicon.

It differs from the best known among current linguistic theories, as for instance Chomsky's deep grammar and its various offspring, which are strictly formalist and syntax-based.

Our approach will be essentially based on the assumption made in LG concerning the fact that a coherent natural language formal description is crucial for developing NLP applications. In this sense, it is worth remembering that the NLP tactic followed by LG plans the structuring of exhaustive and descriptively taxonomic and ontological LRs (i.e. electronic dictionaries, syntactic matrix tables and local grammars).

In order to develop and test such LRs we use NooJ[2], an NLP environment developed by Max Silberztein. NooJ is suitable for developing LRs and also local grammars, in the form of Finite State Automata and Transducers (FSA/FSTs). NooJ processes large *corpora* providing complete results about occurrence outputs, e.g. concordances, matching text units and statistical analyses. This system has been and still is used by a large community, which develops and shares linguistic modules for more than twenty languages. Our analysis and LRs are based on the Italian Linguistic Module, created by Vietri (2014) and maintained by the team of the Laboratory of Computational Linguistics "Maurice Gross" of University of Salerno[3]. Those LRs are composed of simple and compound word electronic dictionaries, inflectional, syntactic and morphological grammars. Therefore, our research enriches the Italian Module developing domain electronic dictionaries and FSA/FSTs used for Term Extraction and semantic annotation tasks.

Thanks to their specific formal characteristics, such LRs have proven to be useful also in the development and implementation of effective Knowledge Management Systems (KMSs) (Marano, 2012).

[2]For more information on NooJ, see *www.nooj-association.org/.*

[3]The Laboratory of Computational Linguistics "Maurice Gross" of University of Salerno, headed by Annibale Elia.

http://labgross.unisa.it/.

2. From Formal Words to Atomic Linguistic Units

The concept of *word* and, consequently, the concept of *word meaning* are problematic to define, due to the fact that both present meanings which depends on contexts of use. For instance, in common use, there exists an ambiguity related to *lexeme* (namely the smallest unit of lexical meaning which constitutes the lexicon of a language) and *linguistic unit* (namely a single word, a part of word or a chain of words which conveys a meaning). Even if both seem to cope with the same purpose, a lexeme is an abstract unit which refers to morphological analysis in Linguistics, while a linguistic unit may be described semantically and/or pragmatically analysing all the sentence contexts in which it may be used.

Therefore, a *word* may be defined in more than a way, which means that we deal with differences derived from the telic essence we associate to the notion.

Among the ways in which we may define a *word,* a fundamental approach is represented by the use of *linguistic* definitions. Such definitions "attempt to characterize the notion of word by illustrating the explanatory role words play or are expected to play in the context of a formal grammar. These approaches often end up splitting the notion of word into a number of more fine-grained and theoretically manageable notions, but still tend to regard 'word' as a term that zeroes in on a scientifically respectable concept (e.g., Di Sciullo & Williams 1987)"[4].

Traditionally, in Linguistics the concept of formal word may be developed according to different layers of analysis, including phonetics, phonology, morphology, orthography, and lexicography. This brings to formulate its notion as follows:

- Lexical, or lexicalized, formal word: it is a composed by a sequence of one or more words, which stands for a unique meaning unit in a given tongue.

[4]Stanford Encyclopedia of Philosophy – Word Meaning. For more information, see: http://plato.stanford.edu/entries/word-meaning/#PhiLan.

- Phonetic formal word: in the speech chain, any sequence of sounds resulting from segmentation, and which we recognize as a lexical or lexicalized word.
- Phonological formal word: it is formed by a sequence of one or more syllables which are characterized by phonological autonomy.
- Morphological formal word: it is composed by a sequence of one or more morphemes which constitute the structure of a lexical or lexicalized word.
- Orthographic formal word: it is composed by a sequence of characters delimited by blank spaces, i.e. it is a formal string bounded by a left and a right blank space.

In our work, lexical words are the main object of analysis, considering our final goal, which aims at studying the relationships likely to occur and be formalized among concepts, signifiers and references.

The achievement of reconstructing the relationship between a word and its meaning leads NLP in growing louder its object of analysis. Indeed, in order to retrieve the semantics of words we may have to cope with more orthographic/lexicalized words than with single and isolated ones. For example, in expressions like 'White House', we may not consider two words in isolation, but we have to confer them a different lexical status. In brief, their properties are not derivable/predictable from/by their component words. For this reason, expressions such as these are referred as MultiWord Expressions (MWEs) or MultiWord Units (MWUs)[5], which stand for contiguous, or not, sequences of simple words which may be computed as a single linguistic unit.

The task of computing MWUs or MWEs attracts (is debated?) the interest of several scholars and researches. The main reason of the interest in such a topic is related to the issue of recognizing groups of words which co-occur in specific contexts. Therefore, the goal is identifying those words, which, when combining with others, become able to carry a different semantic expressiveness and charge than single words.

[5]It is worth stressing that various terms have been largely employed to refer to these expressions, such as multiword lexical items, phraseological unit and fixed expressions.

In LG framework, Gross identifies such lexical groups as summarized in Laporte (2005:2):

> *Gross set up a series of studies on compound lexical units (e.g. Freckleton, 1985; Machonis, 1985), breaking with a long tradition that holds that such phrases are exceptions, worth only of anecdotal remarks. Compound lexical units are phrases described as lexical units (Gross, 1986a), as in:*
>
> (1) *Cell phones have antennas*
>
> *They can be defined by their lack of compositionality and the distributional frozenness of their elements. The findings showed that languages have a large number of compound predicates such as make ends meet, and that compound entries are often more numerous than the simple-word entries for the same part of speech. In French, for instance, Maurice Gross indexed 26,000 verbal idioms (Gross, 1982) and 12,000 adverbial idioms (Gross, 1986b). He devised the notion of local grammars (Gross, 1997) for semi-frozen phrases and for sequences with frozen behaviour inside a specific domain, like cloudy with sunny periods.*
> *The term 'multi-word units' (Glass, Hazen, 1998) is more recent. It groups support-verb constructions, compound lexical units, semi-frozen phrases and collocations.*

Today, most frequentist or probabilistic textual analysis methods, which apply stochastic rules, may collapse on MWU analysis, due for instance to the low frequency of these lexical items in specific texts. In addition, statistical parsing may not appropriately recognize even highly frequent MWUs as single meaning units, consequently losing pieces of information. These and other similar collapses come from the fact that for a MWU as *cell phones*, there is no general probabilistic and/or markovian projection able to stochastically and iteratively predict the occurrence of a word like *phones* after a word like *cell*.

This is because with their unpredictable formation routines[6], and as the result of a continuous interaction between *langue* and *parole,* MWUs are the

[6]For instance, non-compositional MWUs are formed according also to linguistically motivated, manifold, deep-logic contiguous juxtapositions, and not only on the basis of the accreting semantic expansion mechanisms firstly attested by Harris in *From Morpheme to*

clear demonstration that natural language is a complex system. This means that MWUs cannot be stochastically coped with, and must be necessary lexicalized if they present the even slightest non-compositional link among their components. Therefore, we will see that being dictionary-based, our identification and retrieval of Atomic Linguistic Units (ALUs) is founded on a systematic and exhaustive formalization of natural language.

Following Gross' idea, Silberztein (1993) introduces the term ALU which we share and adopt to indicate all elements included in our dictionaries. In other words, ALUs stand for all elements considered essential in order to describe exhaustively the vocabulary of a given natural language.
Indeed, as Silberztein states:

- *les ALU constituent le vocabulaire standard d'une langue, et sont nécessairement en nombre fini: on peut, et on doit les recenser exhaustivement*
- *les ALU ne sont pas analysables meme si elles semblent etre construites à partir d'éléments plus petits par dérivation ou par composition* (Silberztein, 2015:94)[7].

We may define word formations (going from compound terminological words to proverb[8]), having a unique overall meaning, as Atomic Linguistic Units (ALUs). This definition has been borrowed from (Silberztein, 1993), which identifies ALUs as the "smallest elements that make up the sentence, i.e. the non- analysable units of the language", including simple words, affixes, Multi-Word Units and frozen expressions.

In this sense, ALUs hold four types of elements (Silberztein, 2015): simple words, morphemes, compound words and expressions.

Utterance (1946). These mechanisms are useful to fully account only for compositional MWUs, or better for free word groups. Therefore, only these last may have some possibility to be automatically and successfully parsed by means of stochastic routines, which in this case will mainly look like regular expressions, or Markov chains.

[7]- ALUs constitute the vocabulary standard of a given language, they unavoidably represent a finite set: thus, we may and we have to describe them exhaustively.

- ALU are not analysable even if they seem to be built from smaller components through derivation or composition process (Translation by the editor).

[8]For more information, see D'Agostino & Elia (1998).

3. One ALU=One Lexical Entry

As stated by Vietri & Monteleone (2014), we adopt the expression ALUs also to indicate any kind of lemmatizable terminological compound words which, even being very often semantically compositional, can be lemmatized due to their particular non-ambiguous informational content.

In our research, we also identify a fifth kind of elements, which may be described and retrieved as ALUs: NPs which present a restricted semantic expansion. It means that such NPs are formed by a head phrase, generally fixed or semi-fixed, followed by variable elements which belong to specific grammatical categories. These variable elements are characterized by a selection restriction, which is determined by the head phrase – which functions as a predicate – and by the semantic provisions which they represent. In other words, we may define such lexical elements as semi-open NPs, in which the fixed or semi-fixed head defines grammatical and semantic types of all variable elements. This phenomenon is mainly observable inside the lexicons of specific knowledge domains, even if it presents features belonging to both common-usage lexicon and terminology. Indeed, such semi-open NPs are characterized by a variability of non-fixed elements but, at the same time, they are also characterized by a non-ambiguous meaning as a result of the compositional process. The high variability of non-fixed elements is related to the possibility of selecting elements from non-restraint sets of lexical items, the grammatical categories of which are predictable thanks to heads components. On a lexical level, such feature is correlated to the paradigmatic relationship which indicates words belonging to the same POS class. On the other hand, constraints deriving from heads components are associated to the syntagmatic relationship among words.

Thus, for example, in the Archaeological domain, we may observe this phenomenon of semi-open NPs in Coroplastic descriptions, as the following example shows:

(1) *statua di* (statue of) [NPREP]+N
(2) **statua di* (statue of) [NPREP]+A

'Statue of' represents the head of the NP, which determines the type of the element which comes afterwards, that must be a noun (1), and not an adjective (2). Indeed, if the head is composed by a noun followed by a

preposition, like *di* (of), the element which comes afterwards must belong to noun POS. Similarly, the head works as a constraint for the type of noun selected, which means that we have a restricted semantic expansion concerning the semantic type of noun. Thus, the semi-open NP 'statue of' may select a proper noun as 'Silene', or a noun as 'woman', but not a noun as 'table'. On the basis of such selection restrictions, we may identify sets of lexical elements which may co-occur in specific semi-open NPs.

In the following paragraph, we will show how we can recognize and, subsequently, record, such linguistic phenomena through FSA, basing our method on co-occurrence likelihood of elements in semi-open NPs.

3.1 The Archaeological Italian Electronic Dictionary (AIED)

As we have previously stated, LG framework builds electronic dictionaries to describe morphological and grammatical features of lexical entries. Such dictionaries are used as linguistic engines to automatically read and parse texts, therefore to recognize and locate ALUs inside texts.

The Archaeological Italian Electronic Dictionary (AIED) has been developed in order to create the main LRs to be applied by our system during linguistic processing.

Each specific domain has its own terms, which on their turn have specific meanings only when used with reference to that domain. This does not mean that terms may not be polysemic, but only that for a term having two or more meanings, it will be possible to disambiguate usages according to the different specific domains that term belongs to. This is why terminology is used as the basis of KR.

In fact, outlining formalizations, domain terms, which are unambiguous and clear, become useful for conceptualizations.

Sowa (1999) notes that "most fields of science, engineering, business, and law have evolved systems of terminology or nomenclature for naming, classifying, and standardizing their concepts". As well, POS present two levels of representation, which are separated but interlinked: a conceptual-semantic level, pertaining to ontologies, and a syntactic-semantic level, pertaining to sentence production.

As presented in di Buono *et al.* (2014b), we developed the Archaeological Italian Electronic Dictionary (AIED) starting from Thesauri and Guidelines of the

Italian Central Institute for the Catalogue and Documentation (ICCD). In these resources, ICCD provides information about the use of terminology and controlled vocabularies for cataloguers and other professionals. These Thesauri include terms, descriptions and other information needful to objects cataloguing.
ICCD resources are organized in:

1. Object definition dictionary
2. Marble sculptures
3. Metal containers
4. Marble sculptures – Sarcophagi and reliefs
5. Vocabulary of Metals
6. Vocabulary of Glasses
7. Vocabulary of Materials
8. Vocabulary of Mosaic Pavement Works
9. Vocabulary of non-figurative mosaics
10. Vocabulary of Mosaics
11. Vocabulary of Coroplastics.

The Object definition dictionary provides, for each entry, the following different and structured information (see Table 1):

- Broader Term [BT],
- Broader Term Partitive [BTP1],
- Broader Term Partitive [BTP2],
- Narrower Term [NT],
- Narrower Term Partitive [NTP],
- Use [USE],
- Use For [UF].

BT field indicates the general class of the lexical entry, while BTP1 and BTP2 stand for the taxonomy classification. Thus, *amuleto* (amulet) is an element of the BT class *Strumenti, Utensili e Oggetti d'uso* (Tools), which is a general category. In addition, it is also hold in BTP1 class *Amuleti e oggetti per uso cerimoniale, magico e votivo* (Magic & Votive Supplies), which is a specific sub-category of Tools.

The NTP field specifies the lemma, and this helps us to infer that *amuleto* may occur in different compound entries, for instance: *amuleto a forma di anatra* (duck amulet), *amuleto a forma di ariete* (ram amulet) and so on. In order to retrieve such compounds, we apply an FSA-based method (see Paragraph 4.2) which uses inference also to identify and extract taxonomy relationships among words.

UF is a no-preferential lemma (i.e. a variant); this implies that *cornetto* (horn amulet) can stand for *amuleto* (and its specific types), but ICCD guidelines suggest to use the first one.

Table 5.1 – Sample of ICCD Object definition dictionary.

Entry	BT	BTP1	BTP2	NT	NTP	Use	Use For
amuleto	STRUMENTI, UTENSILI E OGGETTI D'USO	AMULETI E OGGETTI PER USO CERIMONIALE, MAGICO E VOTIVO	--	--	a forma di anatra a forma di ariete a forma di colonna a forma di conchiglia a forma di corno a forma di corona bianca a forma di corona rossa a forma di gatto a forma di leone …	--	cornetto

According to our approach, it is necessary to lemmatize all possible variants, including those having even a low frequency use, for example, UF terms from Object definition dictionary.

Therefore, our electronic dictionary, which represents an additional resource to the ICCD ones listed above, includes spelling variants or no-preferential lemmas (UF), i.e.:

(1) *Dinos (+dynos+dèinos*) *con anse ad anello* (ringed-handle dinos+dynos+dèinos).
(2) *Amuleto (+cornetto)* (amulet+horn-shaped pendant)

In AIED entries, also synonyms are included, generally extracted from the UF field, i.e. *lip cup* which stands for *kylix a labbro risparmiato* (spared-lip kylix). The remaining ICCD vocabularies are generally organized into unstructured texts, as the following example, extracted from the Vocabulary of Coroplastics:

> ***Capitello***[9]
> *Elemento posto tra la colonna e la trabeazione. Nell'ambito della decorazione in terracotta il capitello costituisce il rivestimento fittile della struttura lignea.*
> *Un frammento esemplificativo è quello proveniente dall'area sacra di S. Omobono, il quale costituisce il rivestimento del sommoscapo di una colonna e del relativo capitello dorico (…)*[10]*.*

[9]This kind of description is normally used by paper encyclopaedic dictionaries, the linguistic glosses of which, as stated by Gross (1989), are written as unstructured texts. This means that they cannot be automatically imported and/or used for the building and upgrading of LG electronic dictionaries, in so far as they are not written with reference to a formal method of information structuring. Anyway, the information they include can be used inside the descriptive fields of our entries. Actually, parts of the unstructured texts are retrieved by means of FSA in order to create a match between lexical entries and their meaning.

[10]**Capitel**

Element placed between a column and an entablature (a load thrusting). In the field of earthenware decoration, the capitel forms a "fittile" siding in a wooden structure. An exemplifying fragment is the one retrieved in the sacra area of S. Omobono, which constitutes the siding of the higher end of a column and its doric capitel (…).

Thus, in such a vocabulary, descriptions about the ALUs are provided for each entry, in order to offer an accurate model usable to match ALUs and their specific meaning(s). These descriptions deliver data suitable for the development of an encyclopaedic electronic dictionary, even if not all entries present a description. Furthermore, these descriptions differ due to the fact that they are not standardized. In other words, some entries present accurate descriptions and references, while others offer less specific information (e.g., they do not provide object samples). Nevertheless, provided descriptions are very useful to improve our LRs in two ways:

1. They stand for an initial step in the development of an encyclopaedic electronic dictionary which is comparable to existing ones (i.e., WordNet, etc.);
2. They are an additional source from which other ALUs may be extracted.

In order to use also such textual data, for each AIED entry we insert descriptions when they are available in ICCD thesauri, extracting manually just the general ones. Indeed, we resolve to select just general descriptions for levelling out information inserted in our lexical database, while more specific information are skipped. Thus, for example for the entry 'Capitel' we record the description (DEF) as it follows:

> ***Capitello***
> *Elemento posto tra la colonna e la trabeazione. Nell'ambito della decorazione in terracotta il capitello costituisce il rivestimento fittile della struttura lignea.*

Additionally, we employ an automatic procedure to parse such ICCD descriptions, in order to recognize unknown words, that is to say words which are not present in our LRs. Subsequently, unknown words are assessed and inserted in AIED.

Currently AIED is composed of about 11000 ALUs and their related information; indeed, for each entry we indicate (see Table 2):

- Its POS (Category), internal structure and inflectional code (+FLX). These information represent a formal and morphological description. In

fact, the category and the internal structure (Int. Str.) indicate that the given ALU is formally a Noun and is formed by different single elements. In Table 2, the tag "NPREPNPREPN" describes how the given ALU, *dinos con anse ad anello*, is formed (i.e. N stands for Noun and PREP for Preposition). At the same time, the tag "FLX=C610" refers to the ALU number and gender recalling a local grammar in order to generate and recognize correspondent forms (e.g. singular/plural, masculine/feminine).

- Its variants (VAR) and synonyms (SYN), if any;
- The type of link (LINK) (RDF and/or HTML), associated to the linguistic resource. In the following pages, we explain our methodology for integrating Linguistic Linked Open Data (LLOD);
- With reference to a taxonomy, the pertaining knowledge domain (DOM); for our dictionary we have developed a taxonomy, based on ICCD prescriptions, therefore all entries have a terminological and domain label usable for ontologies population.
- The use of domain label subset tags is also previewed for those domain sectors which include specific sub-sectors. This is the case with Archaeological Remains, for which a generic tag «RA1» is used, while more explicit tags are used for Object Type, Subject, Primary Material, Method of Manufacture, and Object Description.
- Finally, we insert a tag referring to ontological classes derived from the ICOM International Committee for Documentation (CIDOC) Conceptual Reference Model (CRM). In AIED we associate to lexical entries the ontology schema provided by CIDOC and compatible with RDF. Actually, the tag CCL allows us to derive definitions and a formal structure for describing the implicit and explicit concepts and relationships used in Cultural Heritage documentation.

Table 5.2 – A selected sample of AIED entries.

Entry	Category	Int. Str.	FLX	VAR	SYN	LINK	DOM	CCL
dinos con anse ad anello	N	NPNPN	C610	dynos/ déinos		RDF	RA1SUOCR	E22
kylix a labbro risparmiato	N	NPNA	C611		lip cup	RDF	RA1SUOCR	E22

Dealing with different type of texts (structured, semi-structured, and unstructured), we have to employ a mixed procedure to ensure the recognition of the largest number possible of lexical entries.

Indeed, AIED has been develop using a two-level procedure to generate/recognize entries:

1. A procedure which is based on manually-built resources.
2. A semi-automatic process which extracts entries and information from external resources.

The first kind of entries are the result of manual lemmatizations, achieved by analysing and recording domain-specific lemmas from ICCD thesauri and guidelines.

The second type of lexical entries are collected through a semi-automatic parsing process, in order to extract unknown words from unstructured texts. This process is achieved through the development of specific FSA, starting from the manually built resources. For example, using a simple FSA which recognizes the capital letter in unknown words, we may extract also Proper and Place Names, with a fairly low percentage of error. Lexical items, identified through such a method, have to be analysed linguistically together with their linguistic behaviours, in order to become entries of our dictionary.

Besides, through a semi-automatic process (see Paragraph 4.2), additional resources have been created, from unstructured texts recorded in existing literature and vocabularies, extracting terms and further information. Thus, for example, from ICCD unstructured data (i.e. the vocabulary of

Coroplastics), we have extracted object descriptions and inserted them in our dictionary as definitions (+DEF).

Consequently, in AIED the entry *amuleto* is associated with the description presented as *Nota d'ambito* (Scope note) in the ICCD object definition dictionary:

> +DEF= *Oggetto di piccole dimensioni a cui sono attribuite superstiziosamente virtù magiche, terapeutiche e protettive favorevoli a chi lo porta con sé [De Mauro]*

According to our methodological framework, free word groups (i.e. compositional non-terminological free word formations) have not been lemmatized in our lexical databases. For such kind of ALUs, i.e. for compositional non-terminological free word formations, we apply the already existing Italian LRs, if any.

Thus, our ALU treatment consists in their recognition and classification by means of formal, morph-grammatical information and terminological tags used to label entries in electronic dictionaries.

Each entry is also given an ontological identification, consisting in tags which send back to the knowledge domain(s) within which entries are commonly used (i.e. in which they have terminological non-ambiguous meanings). Such ontological definition aims at providing an initial semantic description of AIED entries, by means of a domain ontology, and in order to identify class elements and taxonomic labels.

3.2 Semantic Annotation

Semantic annotation represents a key step in our procedure, due to the fact that annotating text requires the capability of matching correctly a natural language formalism and a data model formalism. Indeed, annotation task aims at adding some data to some other data, through the creation of a relationship between annotating data and annotated data.

Actually, as stated in Turney & Pantel (2010), "annotation is the inverse of normalization. Just as different strings of characters may have the same meaning, it also happens that identical strings of characters may have different meanings, depending on the context".

As we will see in Chapter VI, in our system, we present a semantic annotation process which involves both a front-end side and a back-end one.

The main reason for this strategy is related to the possibility of developing a question-answering system, suitable for processing input and output data flows. It means that actually our system analyses:

1. Users' queries, which form the so called front-end side and represent the input data flow and
2. Documents in KBs, the back-end side, namely the output data flow.

Such an annotation process is based on a deep Natural Language Analysis, which means that we perform a linguistic analysis of user's queries and documents in order to annotate them.

The goal of this process is to create a semantic annotation routine describing terms both formally, using the Resource Description Framework (RDF) prescription, and ontologically, using CIDOC CRM. These two types of annotation allow us to extract entities and to infer relationships between them building up a network data.

Particularly, the use of an ontology in the semantic annotation process of LG LRs may ensure knowledge sharing, maintenance of semantic constraints, semantic ambiguities solving, and inferencing on the basis of concept networks. This comes from the fact that ontology-based LRs are likely to incorporate more information than thesauri. In fact, with reference to a thesaurus, an ontology also stores language-independent information and semantic relations.

Thus, during the annotation process we employ two conceptual schemata: one which refers to hierarchy retrieved from ICCD thesauri and another one which represents a standard for cataloguing CH information.

This is why the function of ontological tags differs from the function of classic semantic roles. While the last ones are exogenous – i.e. their nature and quality are governed by predicates by means of co-occurrence and selection restriction rules, ontological tags are endogenous, in the sense that they predicate something about the entity to which they are assigned. That is to say: if we build an RDF triple starting from a given a sentence, we must assume that all syntactic rules are respected; otherwise, the sentence itself would be unacceptable and meaningless. Only such an assumption allows us to verify

that the level of analysis in which ontologies do operate is only the one concerning the relations between concepts.

As for our approach, we can state that predicates are used to define both the semantic roles of the complements and the relations between concepts inside RFD triples.

Therefore, we may divide our semantic annotation task into two subtasks: taxonomic tagging and data representation (ontological tagging). The first task is developed by the means of the taxonomic hierarchy suggested by the ICCD, while the data representation is based on a conceptual reference model for Cultural Heritage contents.

Taxonomic tagging

ICCD thesauri and guidelines indicate how to classify an object, according to taxonomic information presented in BT, BTP1 and BTP2 fields. Thus, in order to reuse such information, we introduce a set of domain labels, which indicate the ICCD hierarchy for a specific knowledge domain. Such set of domain labels is used to annotate AIED entries taxonomically by means of the tag "DOM".

Therefore, starting from a generic RA1 which stands for Archaeological Remains, we use nine specific subfields which refer to the main classes of ICCD classification:

1. Clothing & Personal (RA1AB)
2. Furniture (RA1AR)
3. Building (RA1ED)
4. Lines (RA1MT)
5. Painting (RA1PI)
6. Sculpture (RA1SC)
7. Tools (RA1SUO)
8. Overall Terms (RA1TG)
9. Materials & Techniques (RA1MT).

For each of the first seven main classes, we identify different sub-classes, in order to tag dictionary entries with information related to a specific hierarchical taxonomy. Such a taxonomy, based on the ICCD ones, is composed by 65 classes and sub-classes, which allow to describe items by means of

variable detail levels. Thus, for instance, the primary class *Clothing and Personal* is structured into five sub-classes, as they follow:

1. Fitting (RA1ABAC)
2. Footwear (RA1ABCA)
3. Headgear (RA1ABCO)
4. Jewellery (RA1ABGM)
5. Clothing (RA1ABVE).

On the other hand, some classes present more levels of details, such as the class Building, which, among others, includes a sub-class *Architectural Elements*, split into other two sub-classes, such as *Structural Elements* and *Decorative Elements*. In Figure 1, we present an extract from the taxonomy applied to describe and classify AIED entries.

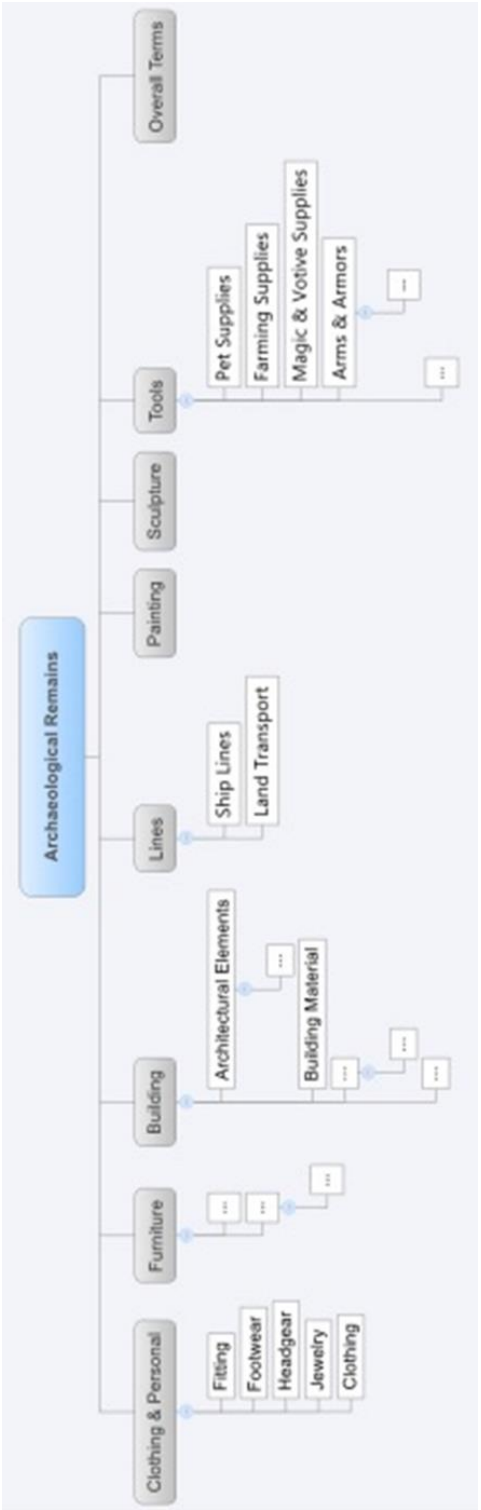


Figure 5.1 – An extract from AIED Taxonomy.

Ontological Tagging

Concerning the standard conceptual model, we rely upon the Conceptual Reference Model (CRM), defined by the Conseil International des Musees (CIDOC), in order to achieve the ontological annotation task. This object-oriented semantic model, compatible with RDF, stands for a domain ontology which may be applied to describe CH objects and the relations among them.

Indeed, as stated in (Doerr, 2003), CIDOC CRM, Version 6.2, May 2015, is composed of 94 classes (which includes sub-classes and super-classes) and 149 unique properties (and sub-properties). In order to guarantee the correct assignment of CH resources to proper entity groups, for each class detailed information are provided, as in the samples which follow:

E19 Physical Object
Subclass of: E18 Physical Thing
Superclass of:
E20 Biological Object
E22 Man-Made Object
Scope note: This class comprises items of a material nature that are units for documentation and have physical boundaries that separate them completely in an objective way from other objects.
The class also includes all aggregates of objects made for functional purposes of whatever kind, independent of physical coherence, such as a set of chessmen. Typically, instances of E19 Physical Object can be moved (if not too heavy).
In some contexts, such objects, except for aggregates, are also called "bona fide objects" (Smith & Varzi, 2000, pp.401-420), i.e. naturally defined objects.
The decision as to what is documented as a complete item, rather than by its parts or components, may be a purely administrative decision or may be a result of the order in which the item was acquired.
Examples:
John Smith
Aphrodite of Milos
the Palace of Knossos
the Cullinan Diamond
Apollo 13 at the time of launch
Properties:

P54 has current permanent location (is current permanent location of): E53 Place
P55 has current location (currently holds): E53 Place
P56 bears feature (is found on): E26 Physical Feature
P57 has number of parts: E60 Number

E53 Place
Subclass of: E1 CRM Entity
Scope note: This class comprises extents in space, in particular on the surface of the earth, in the pure sense of physics: independent from temporal phenomena and matter.
The instances of E53 Place are usually determined by reference to the position of "immobile" objects such as buildings, cities, mountains, rivers, or dedicated geodetic marks. A Place can be determined by combining a frame of reference and a location with respect to this frame. It may be identified by one or more instances of E44 Place Appellation.
It is sometimes argued that instances of E53 Place are best identified by global coordinates or absolute reference systems. However, relative references are often more relevant in the context of cultural documentation and tend to be more precise. In particular, we are often interested in position in relation to large, mobile objects, such as ships. For example, the Place at which Nelson died is known with reference to a large mobile object – H.M.S Victory. A resolution of this Place in terms of absolute coordinates would require knowledge of the movements of the vessel and the precise time of death, either of which may be revised, and the result would lack historical and cultural relevance.
Any object can serve as a frame of reference for E53 Place determination. The model foresees the notion of a "section" of an E19 Physical Object as a valid E53 Place determination.
Examples:
the extent of the UK in the year 2003
the position of the hallmark on the inside of my wedding ring

```
the place referred to in the phrase: "Fish
collected at three miles north of the confluence
of
the Arve and the Rhone"
here -> <-
Properties:
P87 is identified by (identifies): E44 Place
Appellation
P88 consists of (forms part of): E53 Place
P89 falls within (contains): E53 Place
P121 overlaps with: E53 Place
P122 borders with: E53 Place
```

Thus, CIDOC CRM entity classes are described by means of pertaining information about:

- The taxonomic relation among entity classes (i.e., Subclass of);
- A description of class essential properties (i.e., Scope note);
- Sentences which exemplify NL representations used to denote an element belonging to the class;
- The properties which may co-occur with the given entity class. Such information are structured in a way to describe an RDF triple. Thus, as in all triple-based relations, the given entity stands for the subject, the property is the predicate and the second entity class involved corresponds to the object.

As we have previously stated, for each AIED entry, mainly ALUs in the form of NPs, we employ CCL tags in which the class value (i.e., E19) is inserted. This information is suitable for describing the knowledge domain starting from the smallest constituents, namely elements of entity classes. In other words, the use of ontological annotation allows to define a correspondence between NL elements and elements in a specific conceptual schema. Such a match is the basis on which we lay the foundations of our knowledge extraction process suitable for OL and OP tasks.

4. Knowledge Extraction from Unstructured Textual Data

Extracting knowledge from unstructured texts is one of the most challenging tasks in NLP field. Indeed, in unstructured texts, information and

knowledge are encoded in NL sentences, representing concepts which have to be extracted and processed. In other words, concepts, expressed in such a way, require a deeper linguistic analysis, due to the combinatorial features of natural languages, and their internal intricacy. The process of extracting such kind of data requires several steps, aimed at converting texts into a machine-readable format and at identifying the information to be extracted. In order to achieve both the goals, as we have seen, various formal systems have been developed which intend to improve the semantic formalization of NLs. At present, ontology and RDF-based methods are the most promising solutions for conceptual description and modelling of textual data. Indeed, RDF, ontologies and NLs share some characteristics, which makes possible to assume a correspondence among these formalisms. In our opinion, such correspondence is retrievable into RDF predicates, ontological properties and VPs (i.e. the syntactic behaviours of semantic predicates).

Actually, in all of these three formal descriptions, RDF, ontology and natural language, two atomic elements are connected by a central element. Such central element stands for a trigger able to attract the other two elements, which means capable to establish a relationship between them (Figure 2).

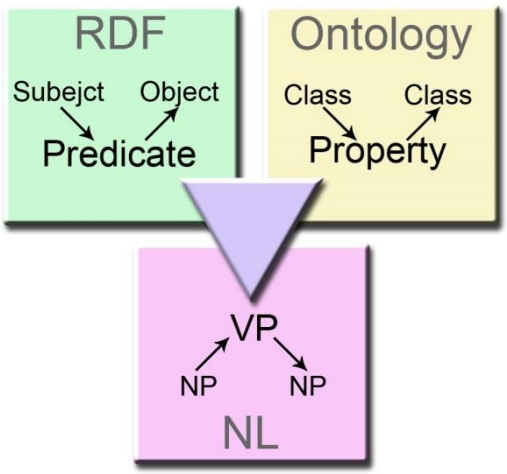


Figure 5.2 – Schema of formal descriptions for RDF, ontology and NL.

Therefore:

- In RDF models the trigger is represented by the predicate;

- In Ontologies, the trigger is represented by the property;
- Finally, in the NL formal descriptions the trigger is represented by the VP.

Thus, in RDF models the trigger is represented by the predicate, in Ontologies by the property and in NL by the VP. On the other hand, atomic elements connected by the trigger are respectively: subject and predicate, two ontological classes and two NPs.

RDF = Subject-*Predicate*-Object
Ontology = Class-*Property*-Class
NL = NP-*VP*-NP

Therefore, these three formal descriptions present a similar logic structure, on the basis of which we may suppose the existence of a parallelism among them. In other words, in such formalisms, we may assume the presence of sets which hold the trigger and two elements involved in the relationship.

Thus, using expressions from the First Order Logic, the parallelism among these three formalisms may be summarized as follows:

RDF:

$P(x,y) \supset x$
$P(x,y) \supset y$

Ontology:

$PR(x,y) \supset x$
$PR(x,y) \supset y$

NL:

$VP(x,y) \supset x$
$VP(x,y) \supset y$

$P(x,y) \supset PR(x,y)$
$PR(x,y) \supset VP(x,y)$
$P(x,y) \supset VP(x,y)$

where P represents the RDF predicate, PR the ontological property, VP the NL VPs and x and y stand for the elements triggered respectively by P, PR and VP.

In other words, assuming the parallelism among such these formal elements, we mean that the VP predicates ontological properties and the NP indicates ontological class elements.

Indeed, as Gross (1984) states, LG considers the simple sentence as the base unit of analysis, which means for LG:

> *A lexicon-grammar is constituted of the elementary sentences of a language. Instead of considering words as basic syntactic units to which grammatical information is attached, we use simple sentences (subject-verb-objects) as dictionary entries. Hence, a full dictionary item is a simple sentence with a description of the corresponding distributional and transformational properties.*

In our research, we focus on one of the three main components which constitute the elements used by LG to formally and taxonomically describe French:

> *- the lexicon-grammar of free sentences, that is, of sentences whose verb imposes selectional restrictions on its subject and complements (e.g. to fall, to eat, to watch)* (Gross, 1984).

Therefore, using properties plus lexical and ontological constraints represents the basis on which we found our linguistic analysis of the deep type as opposite to the shallow one.

4.1 LG Syntactic Tables and Local Grammars

As we have stated, in order to achieve its goals LG framework employs also electronic dictionaries, which are to be intended as lexical and semantic databases. Beyond these ones, other two kinds of resources may be developed to accomplish a complete and exact natural language formalization: syntactic tables and local grammars.

All LG LRs are exactly formalized and fully integrated and interlinked, that is why we may defined them 'encoded and embedded'. Such an expression refers to a type of resources which are built, structured and formalized in order to be used and exploited inside/by a specific NLP environment.

In other words, initially such these resources are the result of accurate observations and analyses, devoted to the recording of all linguistic behaviours of a given language. Typically, linguistic behaviours are the expression of features related to Saussure's dichotomy of *langue* and *parole*, namely a collective/shared normative side and an individual/personal descriptive one. Thus, as mentioned previously citing Gross' 'On the Failure of Generative Grammar' (1979), a given language may be described accounting for its norms and how individuals (i.e. native speakers) use them.

Furthermore, these LRs, developed by means of NooJ, are characterized by the possibility of fitting them in a recursive way, such as in Chinese boxes. It means that they are suitable for describing all kind of languages of Chomsky-Schützenberger hierarchy: rational, algebraic and contextual languages (see Chapter I). Actually, grammar formalisms, introduced in the previous chapter, allow to develop, weak and pure, contextual grammars by means of which they describe contextual languages.

On the other hand, using NooJ in order to develop our LRs allow us to apply two elements:

- *A rational grammar which is also out of context and describes a language larger than contextual ones;*
- *A series of constraints which exclude certain sequences recognized by pure contextual grammars keep only those sequences that truly belong to the desired contextual language* (Silberztein, 2015:228)[11].

LG Syntactic Tables

As Gross explains in *Les bases empiriques de la notion de prédicat sémantique*:

[11] Translation by the editor.

(…)

- *une grammaire G rationnelle ou hors contexte qui définit un langage plus grand que le langage contextuel qu'on veut décrire*

- *une série de contraintes qui excluent certaines séquences reconnues par G pur ne garder que les séquences qui appartiennent véritablement au language contextuel voulu* (Silberztein, 2015:228)

> *Ces études ont abouti à la construction d'un lexique-grammaire représenté par des matrices binaires: à chaque ligne correspond un verbe, à chaque colonne une forme de phrase (e.g. actif, passif, impersonnel). Ces formes sont considérées comme des propriétés du verbe, et à l'intersection d'une ligne et d'une colonne figure un signe « + » quand le verbe entre dans la forme, un « — » dans le cas contraire (cf. Annexe)* (1981:10).

Thus, LG syntactic tables are binary matrixes in which rows correspond to verbs and columns stand for a sentence context (i.e. active, passive and impersonal sentence). In such matrixes information about properties and lexical characteristic of the given verb are inserted. It means that we indicate with a plus + the presence of a characteristic and with a minus – the absence of that characteristic[12].

> *Précisons un point de terminologie: nous appellerons actants syntaxiques les sujet et complément(s) du verbe tels qu'ils sont décrits dans Sy ; nous appelons arguments les variables des prédicats sémantiques. Dans certains exemples, il y a correspondance biunivoque entre actants et arguments, entre phrase simple et prédicat* (1981:9).

Gross defines the subject and the complement of a given verb as syntactic actants, and he also defines arguments the variables of a given semantic predicate. It is worth stressing that just in same cases we have a correspondence between actants and arguments and between simple sentence and predicate.

LG Local Grammars

LG local grammars are developed in the form of FSA/FSTs. An FST is a graph that represents a set of text sequences and associates each recognized sequence to a specific analysis result, also considering their semantics. Text sequences are described in the input part of the FST; the corresponding results are described in the output part of the FST. Conversely, an FSA is a special type of finite-state transducer which does not produce any result (i.e. it has no

[12]In our LG tables, we use 1 or 0 to indicate the presence or the absence of given elements.

output) (Silberztein, 2003). It is typically used to locate morph-syntactic patterns inside corpora, and it extracts matching sequences in order to build indices, concordances, etc. The development of FST/FSA is useful to automatically recognize and tag any kind of text.

When the graph is applied to a text, it recognizes all text accounted for by the sequence of nodes and states. In FSA, words in angle brackets stand for lemma forms and locate all the word forms that are in the same equivalence set as the given word form (generally all inflected, derived forms, or spelling variants of a given lexical entry), the highlighted boxes, in Figure 3, represent a sub-graph (meta-node) that can freely be embedded in more general graphs. Graph embedding allows for reusing sub-graphs in more than one context. At a more theoretical level, it introduces the power of recursion inside grammars.

Sub-graphs may also be used to represent a semantic class and can be encoded in a dictionary with specific semantic features. Electronic dictionaries allow an arbitrary number of semantic features to be represented as tags of lexical entries, and they can also be used in the definition of local grammars.

According to our approach, electronic dictionaries entries are the subject and the object of the RDF triple which allows to recognize predicative relationships in sentence structures. Therefore, through FSA we may rewrite NL sentences directly using RDF schema and OWL, automatically generating the strings while correctly coupling ontologies and ALUs.

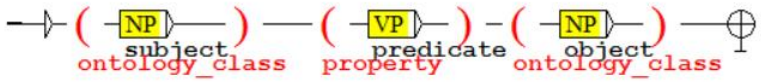


Figure 5.3 – Sample of FSA suitable to match NL sentences, RDF triples and domain ontologies.

All this means that a single FSA/FST can be used to:

- Account for all ALUs referring to a certain class, which means extract and classify terms;
- Account for all declarative sentences of the type “X is a part of Y”, in which X and Y are pre-defined classes, namely construct taxonomic hierarchy among ALUs;
- Allow the matching of POS to a domain ontology in order to extract relation among entities.

Actually, starting from the AIED entries and from their specific tags, we have created additional LRs suitable to formalize the parallelism among RDF schema, a CIDOC CRM ontology and NL.

4.2 Term Extraction and Classification

Term Extraction (TE) approaches require accurate recognition techniques for semantic disambiguation in order to cover several kinds of descriptive data and metadata. Recognizing entities in the Archaeological domain seems to be a more complex task, differing from the Named Entity Recognition (NER) activity in other domains. We share the idea of Ekbal *et al.* (2011): classifying an entity, as a Person, does not comply with the information need related to distinguish if a particular individual was a Painter, a Sculptor, or an Architect, etc. As for the Archaeological Domain, information extraction (IE) outputs must be mainly focused on the rich sets of features which denote and connote any entity likely to be classified as an archaeological object. Besides, such objects, as for instance the Parthenon, are often lexically referred to with capitalized proper nouns.

In this task, the first step to take is the locating of such entities inside texts, that is to say perform automatic entity recognition.

Therefore, in order to achieve TE, we use morph-syntactic information (co-occurrence and selection restriction) to build local grammars. This is due to the fact that local grammars mostly work as a specific tool to cope with special phenomena of language in applications which make use of natural language.

Entities, organizations, persons, locations and time expressions, are also recognized using local grammars. More appropriately, local grammars design is based on the syntactic description which encompasses transformational rules and distributional behaviours. To specify, we build local grammars in form of finite-state transducers (FSTs) and finite-state automata (FSA).

In order to develop our FSA-based system, we apply three types of rules:

- Taxonomic rules (derived from ICCD taxonomy prescriptions);
- Semantic rules (referred to CIDOC Conceptual Reference Model);
- Selection restriction and co-occurrence rules (based on an accurate lexicon formalization).

We analyse domain sentence structures recognizing dependence and co-occurrence rules to develop syntactic FSA. As stated before, syntactic analysis relies essentially on a proper recognition of both verb and noun groups. Since the Italian language presents a high complexity in NPs, we deeply take into account paraphrase constructions and anaphora resolution.

We use resolution techniques based on both eliminative constraints and weighting preferences.

The ontology improves the construction of extraction rules, providing co-occurrence constraints based on properties of each entity.

As we will see, apart from immediate IE, if applied to large corpora, such a procedure can lead to a more complex result, i.e. ontology-based named entity recognition, obtained applying inference during the extraction process, and coupling RDF triples to all the ontologically tagged declarative sentences located during the process of automatic textual analysis.

In order to apply our first technique, we use the grammatical information with which dictionary entries are tagged. As a weighting preference, we also use the information inserted inside the matrix tables created and the concordances made with NooJ.

Therefore, to sum up, TE task can be achieved with a two-step procedure, which includes:

- The creation of ontological matrix tables based on co-occurrence and restriction selection rules of VPs/Properties which govern entity classes.
- The creation and application of FSA/FSTs to specific corpora, in order to extract ALUs and semi-open NPs in which the named entities occur, and in which one or more of their features are explicated or referred to.

Initially, we analyse domain sentence structures recognizing dependence and co-occurrence rules, suitable for identifying entities for RDF triple and associating CIDOC CRM to named entities. Syntactic analysis relies essentially on a proper recognition of both verb and noun groups. In other words, the verb group predicates the ontology properties and the noun group indicates the ontology classes. We use semantic role sets, established on the basis of CIDOC CRM constrains (properties), matched with grammatical and syntactic rules.

Therefore, according to our approach, we match RDF triple with the elements hold in simple sentences. It means that Entity – Domain represents the subject, the property stands for the predicate and the Entity – Range is the object. In order to match recognized NPs with exact CIDOC CRM ontology classes, we develop FSA in which VPs are employed as conditional rules.

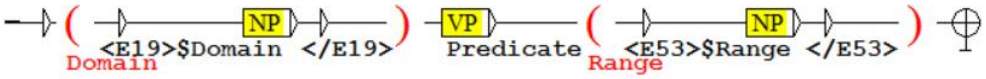


Figure 5.4 – Sample of ontology classes integration in local grammars.

Thus, VPs governs ontology classes, while the FSA (Figure 4) processes the text and tags the NPs through variables, according to the scheme as it follows:

- NP → E19 as Entity – Domain which identifies "Physical Object" class
- NP → E53 as Entity – Range which stands for "Place" class.

The assignment of tags to the NPs is based on the semantic-role set to which the VP belongs. In other words, thanks to the characteristics described in CIDOC CRM, we may identify the specific VP role sets required by such these entity classes.

Thus, the FSA retrieves sentences in which the role pairs "Physical Object" and "Place" are trigged by the predicate *essere situato su* (to be located on) and other VPs belonging to the same role set, such as in:

(1) <E19>*Il Partenone*</E19> *è situato sull'*<E53>*Acropoli di Atene*</E53> (<E19>The Parthenon</E19> is located on the <E53>Acropolis of Athens</E53>).

Applying CIDOC CRM entity classes is suitable to operate one of the two procedures for extracting and classifying terms. The second method is developed on constraints of ICCD taxonomy and is devoted to recognize and extract semi-open NPs. Actually, also in the Archaeological domain, as in many other terminological domains, phrase and sentence structures may present recursive formal structures. Such structures form what in lexicology are called "open series compounds", i.e. lists of compound ALUs having the first two or three items in common. We have defined such ALUs as semi-open NPs, which

means word sequences in which we can identify one or more fixed elements co-occurring with one or more variable ones:

(1) (*palmetta+semipalmetta+rosetta*) <any adjective> <any preposition> +DNUM (*petali+lobi+foglie*) <any adjective> (little palm with (five+six+seven+DNUM) petals).

Such recursive formal structures allow the building of non-deterministic FSA/FSTs, with which it is possible to recognize all the elements of a specific open list as Figure 5.5 shows (di Buono *et al.*, 2013).

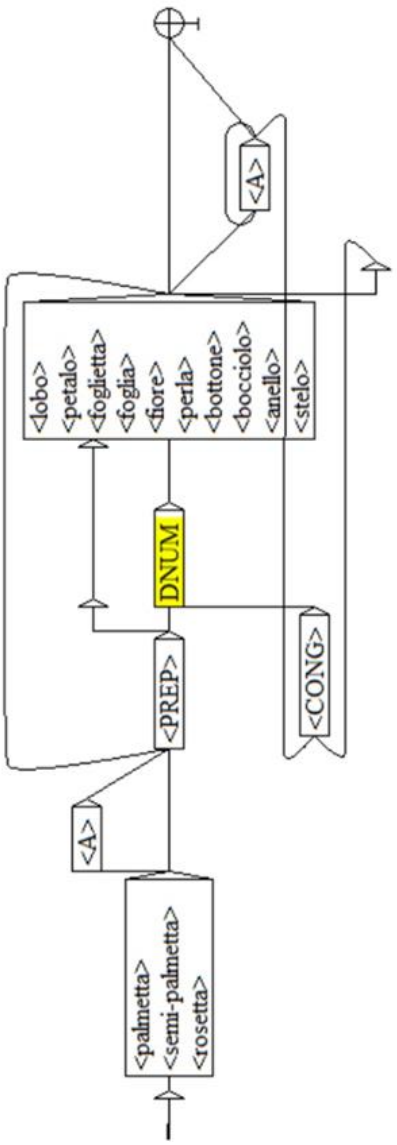


Figure 5.5 – Sample of FSA which recognizes semi-open NPs.

Figure 5.5 shows an automaton which recognizes open series compounds describing a type of decorative feature used in architecture, sculpture, and for earthenware and arms.

In this automaton, the fixed element is represented by *(<palmetta>+<semi palmetta>+<rosetta>)* (<little palm>+<semi little palm>+<rosette>), the variable sequence is composed of both a numerical determiner (<DNUM>) and a set of elements *(<petalo>+<foglia>+<lobo>+…)* (<petal>+<leaf>+<lobe>+…) with its denotative features (<A>). Due to the use of this open series in different subsectors, the inclusion in a certain class of ontology is evinced from the context of the sentence and from the co-occurrences taking place with the verb and all other elements of the sentence. Therefore, in our dictionary, we label entries with a generic tag +DOM=RA1OT[13], while the +CCL tag is assigned using specific syntactic analysis[14].

As for semantics, we observe the presence of semi-open NPs in which the head does not occur in the first position. For example, the open series *frammenti di* (*terracotta*+*anfora*+*laterizi*+N) (fragments of (clay+anphora+bricks+N)), places the heads at the end of the compounds, being *frammenti* used to explicit the notion "N_0 is a part of N_1".

As far as syntactic aspects are concerned, some semi-open NPs, especially referred to Coroplastic description, are sentence reductions in which a present participle construction is used. For instance,

(2) *statua raffigurante Sileno* (statue representing Silenus)

is a reduction (Gross, 1975; Harris, 1976) of the sentence:

(3) a) *Questa statua raffigura Sileno* (This statue represents Silenus)
b) [relative] → *Questa è una statua che raffigura Sileno* (This is a statue which represents Silenus)
c) [pr. part.] → *Questa è una statua raffigurante Sileno* (This is a statue representing Silenus)

[13]RA1OT stands for Archaeological Remains/Overall Terms.

[14]For more information see Ontology Integration in Local Grammars paragraph.

These semi-open NPs, which present sentence reductions, may be retrieved using FSA in which a VP node is inserted (Figure 6).

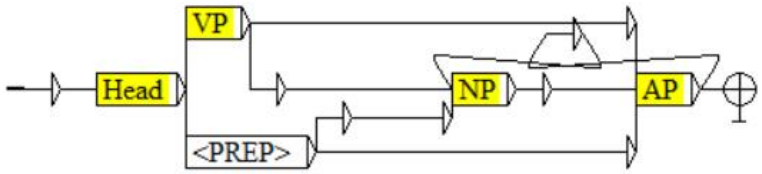


Figure 5.6 – FSA for Coroplastic description semi-open NPs.

Therefore, such an FSA recognizes sentence structures as they follow:

NP*(Head)*+VP+NP
NP*(Head)*+VP+NP+AP
NP*(Head)*+VP+AP+NP
NP*(Head)*+PREP+NP
NP*(Head)*+PREP+NP+AP
NP*(Head)*+PREP+AP+NP.

In the previous sample, the NP which stands for the head of semi-open NPs is composed by a group of non-restricted nouns related to Coroplastics. It means that in such a group we insert nouns as statue, bust, figure and so on (Figure 6). In order to create these semantic groups, we firstly employ information stored in the tag 'DOM', which refers to domain taxonomic hierarchy. In other words, in the sample, our first selection restriction is constrained by the tag value 'RA1SC' which indicates 'Sculpture' class in the taxonomy. Therefore, we extract all ALUs, labelled with Ra1SC tag, from AIED through a semi-automatic method. Consequently, a manual procedure is employed to identify nouns which fit to the meaningful sentence context.

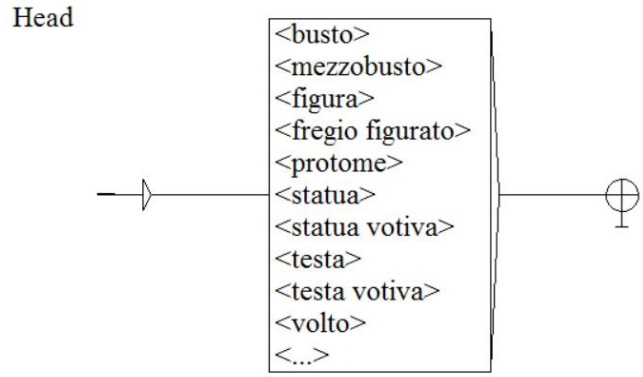


Figure 5.7 – Head sub-graph in semi-open NPs for Coroplastic description.

In compounds containing present participle forms (Figure 6), semantic features can be identified using local grammars built on specific verb classes (semantic predicate sets) (Figure 8); in such cases, co-occurrence restrictions can be described in terms of lexical forms and syntactic structures.

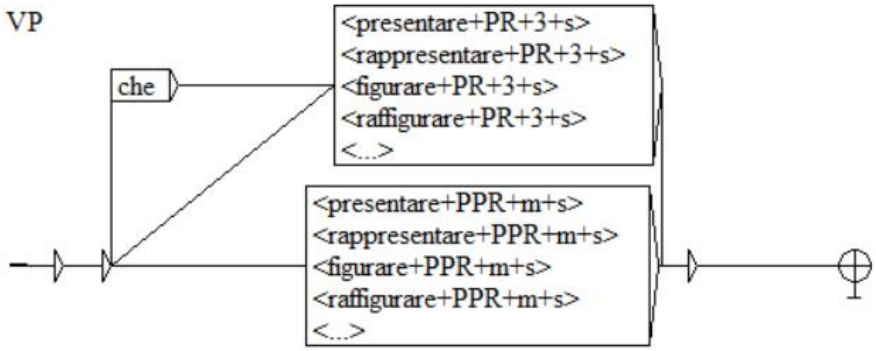


Figure 5.8 – VP sub-graph in semi-open NPs for Coroplastic description.

Figure 8 shows the sub-graph for the VP in Coroplastic descriptions; inside it, we did not use the specific semantic set, descriptive predicates, in order to put into evidence elements extracted from the verbal classes (i.e., 20A and 47B[15]). We also employ grammatical and syntactic constraints referred to tense and number of VP; thus, we select just present 3rd persons singular and plural (sample 3a and 3b) and present participle (sample 3c).

[15]Classes refer to Italian Lexicon-Grammar Tables, available at http://dsc.unisa.it/composti/tavole/combo/tavole.asp.

The NP e AP nodes are suitable to recognize the variable part in such a semi-open NPs. They refer to sets of nouns which may co-occur in such construction on the basis of a selection restriction governed by the head. For example, the NP (Figure 6) holds, among others, constraints structured as <N+Um> and <N+Anl> which selects nouns labelled with 'Human' and 'Animal' tags[16].

Due to the complexity of Coroplastic descriptions, sub-graphs presents many recursive nodes, especially <N>. The structure allows to retrieve a very large amount of expressions and ALUs among those present in corpora analysed.

The main formal structures of semi-open NPs, recorded in AIED are:

- Noun*(Head)*+Preposition+Noun+Preposition+Noun (NPREPNPREPN), i.e. *fibula ad arco a coste* (ribbed-arch fibula), in which the fixed component is represented by *fibula* (fibula);
- Noun*(Head)*+Preposition+Noun+Adjective (NPREPNA), i.e. *anello a capi ritorti* (twisted-heads ring), the head is represented by *anello* (ring);
- Noun*(Head)*+Preposition+Noun+Adjective+Adjective (NPREPNAA), i.e. *punta a foglia larga ovale* (oval broadleaf point).

4.3 Taxonomic Relation Construction

Taxonomy construction represent a task useful in different applications, as for instance question answering and document clustering. At any rate, several handcrafted resources are available, for example WordNet, OpenCyc and Freebase, and the demand for constructing taxonomies for new domains is still growing.

Taxonomy construction is based on the preliminary assignment of domain tags (DOM) which designate the ICCD hierarchy. Thus, we employ such tags to develop FSA suitable for recognizing and extracting taxonomical information.

[16]<N+Um> and <N+Anl> tags refer to the labelling system developed by the Linguistic group of the Laboratory 'Maurice Gross' - University of Salerno.

The process applies an If-Then conditional statement based on ALU co-occurrence evaluated in the context of simple sentences. It means that we draw upon two kinds of constraints: grammatical and taxonomic restrictions.

Similarity between two ALUs is given by the likelihood of occurring in a certain position in sentence contexts. Indeed, starting from simple sentence structure, sequentially formed by a first NP, a VP and a second NP, we may use DOM tags for constraining the IS-A relationship among elements.

Using variables in FSA, we may distinguish the IS-A hierarchical relationship through the domain labels applied to dictionary entries (Figure 5.9).

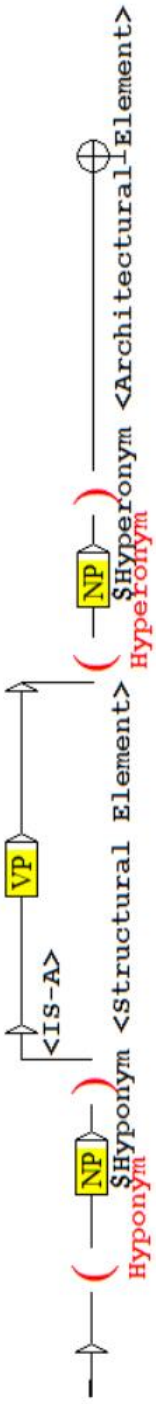


Figure 5.9 – Sample of FSA to extract IS-A relations.

In other words, DOM tags are suitable for recognizing specific sets of AIED entries, which allow to describe the relations among NPs indicating the taxonomic hierarchy.

In Figure 9 the first NP node is used to recognize all entries which are labelled with RA1EDEAES (Structural Element), while the second NP node identifies entries tagged with RA1EDEA (Architectural Elements). The relation between the first and the second NP is established on the basis of semantic predicates of VPs, such as in the sample it follows:

(1) *Un fregio dorico decora il tetto* (A Doric frieze decorates the roof).

The analysis result, computed by the sample FSA, is the following one:

(1a) *Fregio dorico* <Structural Element> <IS-A> *tetto* <Architectural Element>.

In some cases the relation hyponym-hyperonym has a different distribution in sentence contexts, thus, we also take in account passive constructions. In addition, we indicate if a verb allows active/passive constructions and/or transformations, in order to recognize entities also when analysing transformed active declarative sentences. We use local contextual filtering rules, manually constructed.

4.4 Relation/Property Extraction

As we have previously seen, relation/property extraction refers to a task which aims at recognizing non-taxonomic relation among elements. In other words, we have to discover and select:

1. Verbs which express such relationships in a specific knowledge domain;
2. Entities which are involved in this relation as representative elements of concepts.

According to Gross, these relations may be analysed on the basis of three kinds of considerations:

1. Morphological consideration: to some extent, the two categories verbs and adjectives are the predicates, so that the names will instead be arguments;
2. Syntactic considerations: it is found that the number of arguments varies with verbs (which thus appear as functions of several variables). Furthermore, the correspondence between syntactic actants (subject, supplements) and arguments, i.e. the rules of interpretation, involves syntactic actants marking parameters: the order of words, the preposition, the case, etc.;
3. Semantic considerations: given a word, each of its actants has a particular selection from the group names. Nevertheless, this selection varies with each verb, it is what determines the meaning of the verb (Harris, 1952) (Gross, 1981:11)[17].

Therefore, we develop our matrix tables starting from the semantic role sets established on the basis of CIDOC CRM properties, and matched with grammatical and syntactic rules. In other words, CIDOC CRM properties stand for the semantic constraints, while the morphological and syntactic constraints are related to language uses and formalizations.

Actually, the co-occurrence, distributional and combinatory rules which govern our matrixes are defined assuming a correspondence among RDF

[17]Translation by the editor.

Cette notation est traditionnellement justifiée par des considérations de trois types:

1. *morphologiques : dans une certaine mesure, les deux catégories verbes et adjectifs sont les prédicats, alors que les noms seront plutôt les arguments;*
2. *syntaxiques : on constate que le nombre des arguments varie avec les verbes (qui apparaissent donc comme des fonctions à plusieurs variables):*

   Les gâteaux moisissent = Moisir (p)
   Max montre un gâteau à Luc = Montrer (p, q, r)

   De plus, la correspondance entre les actants syntaxiques (sujet, compléments) et les arguments, autrement dit les règles d'interprétation, mettent en jeu les paramètres syntaxiques du marquage des actants : l'ordre des mots, la préposition, le cas, etc. ;
3. *sémantiques : étant donné un verbe, chacun de ses actants a une selection particulière dans l'ensemble des noms. Or cette sélection varie avec chaque verbe, c'est elle qui détermine le sens du verbe (Harris 1952).*

predicates, ontology properties and VPs (i.e. the syntactic behaviours of semantic predicates).

The matrix lists a certain number of verbal entries and a specific number of distributional and syntactic properties.

CCL label and grammatical information, with which dictionary entries are tagged, are the basis on which we develop role set matrixes. Such matrixes are useful to identify predicate-argument structures related to sentence contexts and consequently to achieve the semantic annotation process. Context information inserted inside the matrix tables together with NooJ concordances are employed as weighting preferences.

Thus, we have developed syntactic tables for each CIDOC CRM property recording the distribution and co-occurrence in sentence contexts.

These matrix tables are developed analysing semantic role sets established on the basis of CIDOC CRM constraints (properties) matched with grammatical and syntactic rules. In addition, they indicate if a verb allows active/passive constructions, in order to recognize entities also when analysing transformed active declarative sentences. Indeed, almost all CIDOC CRM properties present a passive construction, except:

- P3 has note
- P57 has number of parts
- P79 beginning is qualified by
- P80 end is qualified by
- P81 ongoing throughout
- P82 at sometime within
- P90 has value
- P114 is equal in time to
- P121 overlaps with
- P122 borders with
- P132 overlaps with
- P133 is separated from
- P139 has alternative form

Table 5.3 – Sample of ontological LG matrix table.

NP E19				VP P54/P55				NP E53			
(E+DET)	A	NA	(Npr+N)	NPREPN	V	AVV	PREP	PREPA	DET	NPREPN	NA
1	0	0	1	0	sorgere	0	0	1	0	1	0
1	0	0	1	0	comparire	0	1	0	1	1	0
1	1	1	0	0	porsi/erigersi/trovarsi	0	1	0	1	0	0
1	0	0	1	0	essere posto/ essere eretto	1	0	1	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...

Such matrix table results from the intersection of LG Italian tables[18] and CIDOC CRM constraints. Indeed, starting from a property (i.e., P54) we retrieve table classes (i.e., CL 57 and CL 8) which refer to VPs identified for the given property.

Table 5.3 shows the matrix for P54 and P55 properties, which stand for "has current location (currently holds)" and "has current permanent location". These properties present the E19 class ("Physical Object") as Entity – Domain and the E53 ("Place") class as Entity – Range, as showed in the following definitions taken from CIDCO CRM.

[18]http://dsc.unisa.it/composti/tavole/combo/tavole.asp.

P54 has current permanent location (is current permanent location of)
Domain: E19 Physical Object
Range: E53 Place
Quantification: many to one (0,1:0,n)

Scope note: This property records the foreseen permanent location of an instance of E19 Physical Object at the time of validity of the record or database containing the statement that uses this property.

P54 has current permanent location (is current permanent location of) is similar to *P55 has current location (currently holds).* However, it indicates the E53 Place currently reserved for an object, such as the permanent storage location or a permanent exhibit location. The object may be temporarily removed from the permanent location, for example when used in temporary exhibitions or loaned to another institution. The object may never actually be located at its permanent location.

Examples:

- silver cup 232 (E22) *has current permanent location* Shelf 3.1, Store 2, Museum of Oxford (E53)

In First Order Logic:

P54(x,y) ⊃ E19(x)
P54(x,y) ⊃ E53(y)

P55 has current location (currently holds)
Domain: E19 Physical Object
Range: E53 Place
Subproperty of: E18 Physical Thing. P53 has former or current location (is former or current location of): E53 Place
Quantification: many to one (0,1:0,n)

Scope note: This property records the location of an E19 Physical Object at the time of validity of the record or database containing the statement that uses this property.

This property is a specialisation of *P53 has former or current location (is former or current location of).* It indicates that the E53 Place associated with the E19 Physical Object is the current location of the object. The property does not allow any indication of how long the Object has been at the current location.
P55 has current location (currently holds) is a shortcut. A more detailed representation can make use of the fully developed (i.e. indirect) path from E19 Physical Object through *P25 moved (moved by)*, E9 Move *P26 moved to (was destination of)* to E53 Place if and only if this Move is the most recent.

Examples:

- silver cup 232 (E22) *has current location* Display cabinet 23, Room 4, British Museum (E53)

In First Order Logic:

P55(x,y) ⊃ E19(x)
P55(x,y) ⊃ E53(y)
P55(x,y) ⊃ P53(x,y)

Italian LG tables which identify P54 and P55 properties are the Verb Classes 57 and 8, defined as it follows:

CLASSE 57: N_0 V a V_{inf} W (50 usi verbali)

Gli usi verbali di questa classe hanno una struttura N_0 V Loc N_1, in cui N0 è generalmente N_{um}, cioè un sostantivo "umano" e al complemento locativo Loc N_1 corrisponde un'infinitiva in A che risponde alla domanda DOVE V N_0? Sono verbi tradizionalmente definiti "di movimento" o "locativi", come nelle frasi:

Max va a nuotare
Eva corre a comprare il pane

CLASSE 8: N_0 V Loc(st) N_1 (24 usi verbali)

Gli usi verbali di questa classe entrano in una struttura in cui N_0 entra in un rapporto di localizzazione "statica" (indicata da V) rispetto all'argomento locativo (N_1) introdotto da una preposizione locativa quale "in", "su", "da","a", ec., come nelle frasi:

Lea abita in via Cilea
La penna è sul tavolo
Leo resta in ufficio[19].

[19]CLASS 57: N_0 V to V_{inf} W (50 verbal uses)

Verb uses of this class have a structure N_0 V Loc N_1, where N_0 is generally an N_{um}, which means a noun referred to a Human noun, and Loc N_1 corresponds to an infinitive phrase in A which answers the question WHERE V N_0? Verbs are traditionally defined as "movement" or "rental", as in the phrases:

Max goes swimming
Eva runs to buy bread.

CLASS 8: N_0 V-$Loc_{(st)}$ N_1 (24 verbal uses)

Verb uses of this class enter into a structure in which N_0 is into a relationship of localization "static" (indicated by V) concerning the subject locative (N_1) introduced by a rental-value preposition which may be "in", "on", "from "," to ", and so on, as in the phrases:

Lea lives in Via Cilea
The pen is on the table
Leo is still in the office.

From these verb classes, we extract just VPs which may co-occur with specific ontological classes, which means that for instance, we do not extract VPs presenting N_{um} in the position of N_0.

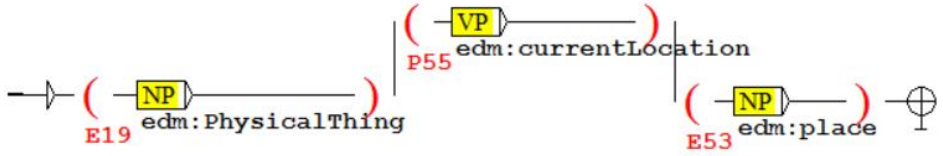


Figure 5.10 – Sample of FSA for RDF/EDM schema.

Figure 5.10 gives a sample of FSA/FST variables associated to and applied with an RDF schema for the following sentence:

(1) *Il Partenone* (subject) *sorge sulla* (predicate) *Acropoli di Atene* (object) (The Parthenon is located on the Acropolis of Athens).

Therefore, we use FSA/FSTs to identify classes and properties for RDF subjects, objects and predicates to which the SKOS concept scheme will be associated. To each instance, we add a meaningful relationship with other instances in terms of RDF triple, in which the predicate is the descriptor annotated by means of a Uniform Resource Identifier (URI) extracted from Dublin Core Metadata Model.

Such a SKOS/RDF concept scheme will be expanded by means of new instances or associative links/relationships, i.e. by adding URIs dealing with concepts and associative relationships among such concepts. This procedure will grant a coherent semantic expansion suitable to improve natural language query effectiveness.

Thus, we may identify semantic groups among properties, which also have, on a lexical level, shared characteristics. These characteristics derive from lexical features related to specific semantic predicates, which together with the labelled dataset, represent the constraints during the recognition process.

In Figure 5.10, we indicate specific POS in the form of sub-graphs, identified as NP, in order to consider the high variability of the lexical class and not of the single form belonging to the class. The NP contains all information, listed in role set matrixes, which can co-occur for that verb.

Besides, in Figure 5.10, we used variables to recognize all instances, i.e. ALUs, which could be included in specific classes, on the basis of the selected

property. When the FSA recognizes the VP, it assigns the related property on the basis of which classes are assigned to NPs.

In order words, before the assignment of ontological classes to NPs, such an FSA recognizes and selects the correct property which the verb predicates. Thus, if the verb belongs to the semantic role set related to the Property 54, the FSA tags the first NP with E19 class, and the second NP with E53 class. Otherwise, if the verb belongs to the semantic role set related to the Property 13, the first NP will be annotate with E6 tag and the second one with E19 one.

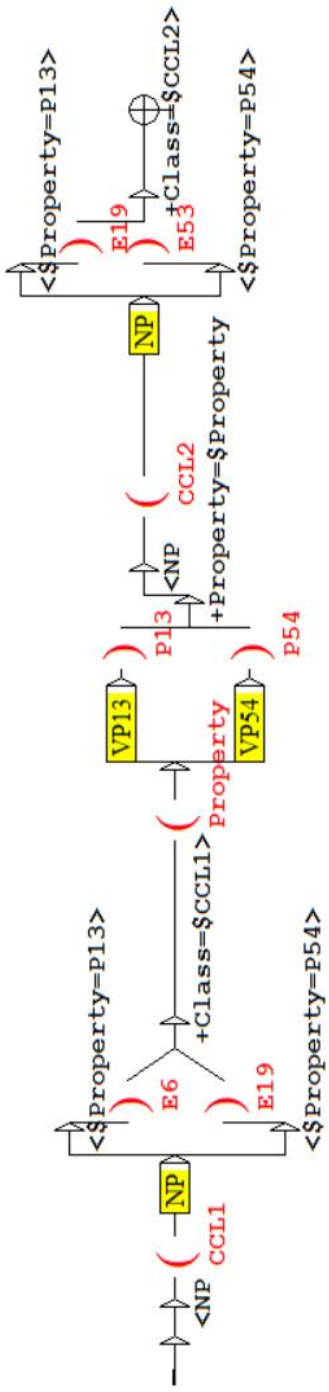


Figure 5.11 – Sample of FSA with variables and CCL tags.

After the phase of text processing, the result is as follow:

(1) *Il lungo fregio dorico posto lungo le pareti esterne della cella*, The long Doric frieze placed along the exterior walls of the cell[20].

<S> <NG> Il lungo <subject> <E19> <N+A> fregio dorico </N+A> </E19> </subject> </NG> <VG> <P54> <V+PREP> posto lungo </V+PREP> </P54> </VG> le <NG> <object> <E53> <N+A+PREP+N> pareti esterne della cella </N+A+PREP+N> </E53> </object> </NG> </S>[21].

<S> <NG> The long <subject> <E19> <N+A> doric frieze </N+A> </E19> </subject> </NG> <VG> <P54> <V+PREP> placed along </V+PREP> </P54> </VG> the <NG> <object> <E53> <N+A+PREP+N> exterior walls of the cell </N+A+PREP+N> </E53> </object> </NG> </S>[22].

[20]Sample recognized processing *Partenone* entry from Italian Wikipedia.

[21]Spaces between tags have been added in order to facilitate the reading.

[22]Spaces between tags have been added in order to facilitate the reading.

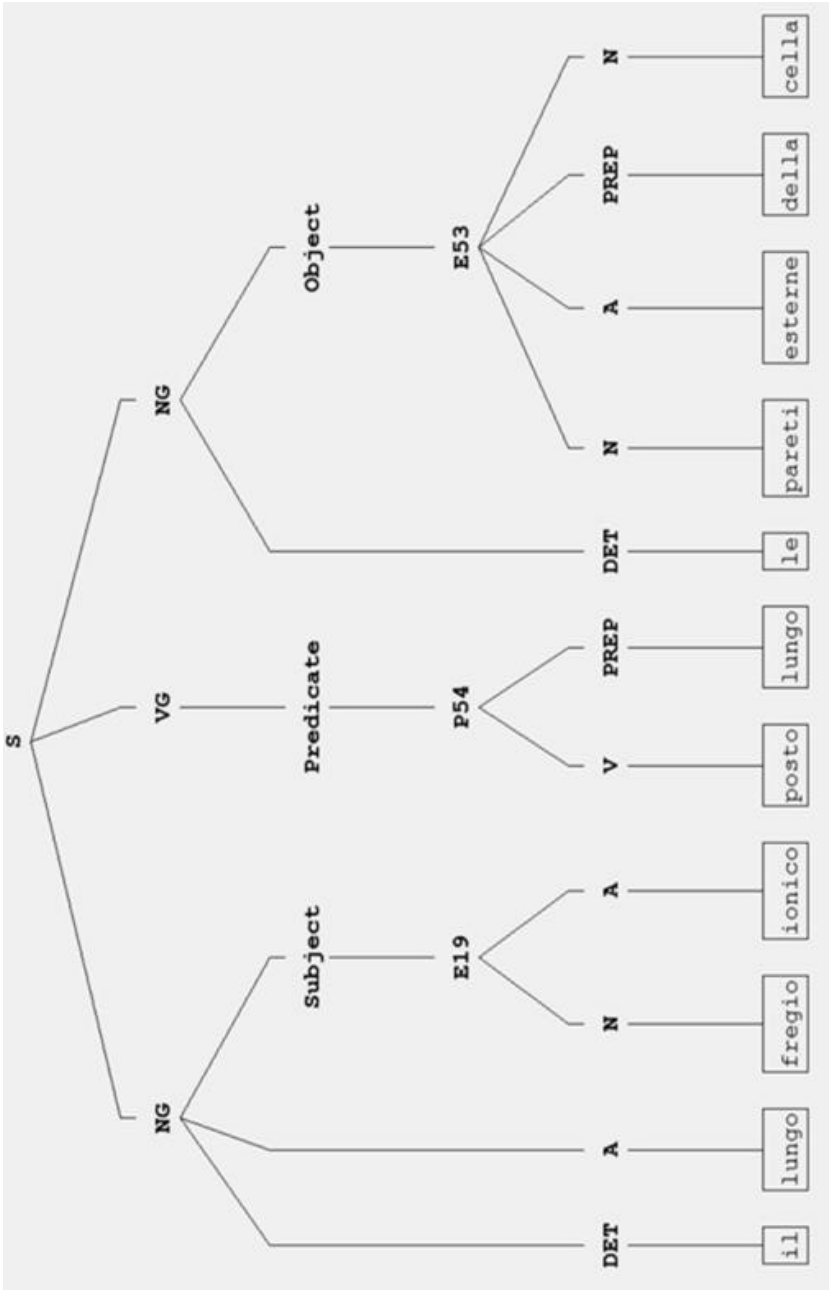


Figure 5.12 – Sample of Syntactic Tree.

Figure 5.12 presents the syntactic tree for the sample sentence, in which Subject, Predicate and Object represent the RDF Triple elements, which are connected to CIDOC CRM Entities and Property.
Therefore, in the sentence

(1) *Il Partenone fu costruito sull'Acropoli di Atene per volontà di Pericle*
(The Parthenon was built on the Acropolis of Athens at the behest of Pericles)

we do identify three entities, tagging *Pericle* with E21 (Person), but the RDF triple will take into account only the property which links the *Partenone* to the *Acropoli di Atene*, property which is explicated by the VP *fu costruito sulla*.

While taking for granted syntax correctness and semantic roles appropriateness, we use ontologies to sever sequences having an identical formal structure, as for instance *da Pericle* and *sull'Acropoli*, which are both formed by a preposition and a noun, but which have different semantic roles and ontological values.

4.5 Linguistic Linked Open Data (LLOD) Integration

Linking LRs with other resources seems a crucial step in order to combine information from different knowledge sources. Indeed, according to Chiarcos *et al.* (2013), "linking to central terminology repositories facilitates conceptual interoperability".

In order to achieve a semantic expansion combining information from different knowledge sources, we also integrate and link our dictionary entries with LLOD.

As we have argued in the previous chapter, the LLOD is a project developed by the Open Linguistics Working Group (OLWG). It aims at creating a representation formalism for corpora in Resource Description Framework/Web Ontology Language (RDF/OWL). The initiative intends to link LRs, represented in RDF, with the resources available in the Linked Open Data (LOD) cloud.

According to the LOD paradigm, Web resources have to present a Uniform Resource Identifier (URI) for entities to which they refer to, and to include links to other resources. "Linking to central terminology repositories

facilitates conceptual interoperability", allowing at creating, through URIs, dynamic connecting between resources (Chiarcos *et al.,* 2013b).

To achieve this goal, the Open Linguistics Working Group (OLWG) developed the Linguistic Linked Open Data (LLOD) project. The initiative intends to link LRs, represented according to the RDF format, with the resources available in the LOD cloud. According to the LOD paradigm (Berners-Lee *et al.*, 2006), Web resources have to present a URI for entities to which they refer to, and to include links to other resources.

The LLOD project aims to create a representation formalism for corpora in RDF/OWL. The LLOD goal is not only to provide LRs in an interoperability way, but also to use an open license. Benefits of LLOD are also identified in linking through URIs, federation, dynamic linking between resources (Chiarcos *et al.*, 2013b).

Data structured in RDF format can be queried by means of the SPARQL language. Indeed, if RDF triples represent a set of relationship among resources, than SPARQL queries are the patterns for these relationships.

One of the most relevant LLOD resources is stored in and presented by DBpedia. DBpedia is a sample of large Linked Datasets, which offers Wikipedia information in RDF format and incorporates other Web datasets.

Therefore, we have referred and will refer to DBpedia Italian datasets to integrate our LRs with LLOD. DBpedia Italian is an open project developed and maintained by the Web of Data research unit of Fondazione Bruno Kessler (FBK). According to Linked Data prescriptions, URI schema is structured as in Table 4.

Table 5.4 – Sample of URI schema.

Resource URI
http://it.dbpedia.org/resource/ordine_dorico
HTML representation
http://it.dbpedia.org/page/ordine_dorico
Machine-readable resource representation
http://it.dbpedia.org/data/ordine_dorico.{ rdf | n3 | json | ntriples }

In order to reuse such prescriptions, we adopt a Finite State Transducer-based system that merges specific matching URIs with electronic dictionary entries.

When we apply the transducer to dictionary entries tagged with "LINK=RDF", NooJ generates a new string in which the resource URI is placed before the original entry. In this way, the transducer enriches all entries of our electronic dictionary with DBPedia resources.

Resulting strings may be used to automatically read texts by means of Web browsers and/or RDF environments/routines. When the generated string is processed by a Web Browser, it will produce a link to the HTML representation. Otherwise, when the header "HTTP Accept:" of the query is produced by a RDF-based application, it will result in a link to the machine-readable resource representation.

The most relevant LLOD resources are stored in and presented by DBPedia (www.dbpedia.org). DBPedia is a sample of large Linked Datasets, which offers Wikipedia information in RDF format and incorporates other Web datasets. We refer to DBPedia Italian datasets[23] to integrate our LRs with LLOD. In order to reuse such prescriptions, we adopt a Finite State Automaton system which merge specific URIs with electronic dictionary entries.

We use an inflectional grammar in order to add the DBpedia/resource link to AIED entries (Figure 13).

Figure 5.13 - Sample of FSA used for generating URIs.

The transducer generates a new string in which the resource URI is placed before the original entry. In this way, the transducer enriches all entries of our electronic dictionary with DBPedia resources. For instance, the result given by the transducer for the compound *Ordine dorico* (Doric order) is the following string:

[23]DBPedia Italian is an open project developed and maintained by the Web of Data research unit of Fondazione Bruno Kessler.

```
# This dictionary was automatically built from archaeology.dic
#
#use rdf.nof
#
<http://it.dbpedia.org/resource/Ordine_dorico>,Ordine dorico,N+FLX=LLOD_C+URI
```

Figure 5.14 - Sample of dictionary output with URIs.

In order to apply also the standard inflectional grammar to entries (for singular and plural forms), we use a normalization process. Such a normalization process also allows to invert the order in dictionary strings and so to have in the first position the lemma and not the link.

Resulting strings may be used to automatically read text by means of Web browsers and/or RDF environments/routines. When the generated string is processed by a Web Browser, it will generate a link to the HTML representation.

Otherwise, when the header "HTTP Accept:" of the query is produced by a RDF-based application, it will produce a link to the machine-readable representation.

In the following chapter, we will show how such LRs developed by means of ontological constraints and LG framework may represent the main resources in the workflow of our environment for semantic knowledge extraction and representation. Specifically, we will demonstrate how crucial is the possibility to process text using all the tags previously presented, be they morphological, syntactic, terminological or ontological.

VI – ENDPOINT FOR SEMANTIC KNOWLEDGE (ESK)

Any sufficiently advanced technology is indistinguishable from magic.
Arthur C. Clarke

In this chapter, we will present the system workflow we intend develop in order to integrate our LRs in an environment suitable for a semantic search engine, called Endpoint for Semantic Knowledge (ESK). ESK will be structured as a SPARQL endpoint, which will be applying a deep semantic analysis, based on the development of a matching process between a set of machine semantic formalisms and a set of NL sentences.
As reported in the Web site for the Semantic Web,

> *A SPARQL endpoint enables users (human or other) to query a knowledge base via the SPARQL language. Results are typically returned in one or more machine-processable formats. Therefore, a SPARQL endpoint is mostly conceived as a machine-friendly interface towards a knowledge base. Both the formulation of the queries and the human-readable presentation of the results should typically be implemented by the calling software, and not be done manually by human users*[1].

Different research groups are nowadays developing various endpoint systems, as for instance Virtuoso[2], which is devoted to run queries against online KBs, mainly DBpedia[3].

[1]Source: http://semanticweb.org/wiki/SPARQL_endpoint.html.

[2]http://dbpedia.org/sparql.

[3]Other samples of endpoints, which allow to access DBpedia KB, are indicated in http://www.w3.org/wiki/SparqlEndpoints.

Anyway, due to the spread of several KBs, different endpoints are available. For more information on current alive SPARQL endpoints, see:

http://www.w3.org/wiki/SparqlEndpoints.

Virtuoso offers a SPARQL Service Endpoint structured as a Server-hybrid Web Application, namely a Universal Server, which provides SQL, XML, and RDF data management in a single multithreaded server process.

Generally speaking, such endpoints allow to access KBs by means of an interface from which it is possible to run SPARQL queries, as it happens with query editors. In other words, endpoints do not process NL queries, but require a structured query, which means that they handle just the information stored as RDF triples into KBs.

Therefore, ESK proposes an interface in which users may insert NL queries and run them again a KB. Furthermore, ESK is set to offer a tool for on-line processing of unstructured texts and Web pages, in order to generate semantically tagged documents. Such a system has to be based on a deep linguistic analysis, which aims at processing both NL and structured information.

1. Indexing Information

The process of information and knowledge retrieval and extraction is composed of different steps, which are summarized by Teufel (2014):

> *finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).*

Such definition allows us to focus on the main characteristics of the process we intend to improve with ESK**.** Actually, three main elements are the core of the process ESK will be accomplishing, that is:

1. Documents, namely unstructured texts,
2. Information needs, expressed through queries,
3. Digital large collections[4], for instance KBs.

[4]It is worth noticing that the expression 'digital large collections' is a blanket term which may refer to different kinds of resources. In this dissertation, our only acceptation is "structured and formalized collections of (un/semi-)structured texts, namely KBs".

We know that KP aims at extracting relevant documents from KBs, processing user's queries and indexing relevant documents, which may correspond to the information required.

Actually, IR and KE tasks are accomplished by means of a very important indexing process[5], which guarantees a fast and accurate KP. Generally speaking, automated indexing is considered faster and cheaper than the one which may be achieved by means of manually built or ruled-based systems, even if its results do not seem to have a high level of accuracy. Automated indexing is achieved using stochastic algorithms, even if this method entails:

- Low precision, due to the indexing of sentences and of common Atomic Linguistic Units (ALUs), which are chunked down to their single tokens;
- Low recall, caused by the presence of synonyms.
- Generic results arising from the use of too board or too narrow terms (Hjorland, 2007).

Usually, rule-less IR systems are based on invert text index, namely an index data structure, which starting from the content processed, and with reference to its location, creates and stores a map in a database file, or in a document or a set of documents.

Most traditional rule-less IR systems process each document separately, to retrieve terms in free-text query, which means that they do not compare results provided from different sources.

Such lack of integration in results causes an overlapping and a decreasing in the positive predictive value[6], due to the fact that the shared content are indexed several times. Various approaches have been proposed to overcome this boundary, increasing recall and precision in results.

Furthermore, indexing process holds some problems related to information fragmentation and the growing complexity of KBs. Most

[5]"Indexing connotes the processes of creating an index. It is derived from the Latin root "índicare," to point or indicate (Chakraborty and Chakrabarti 1983). Its current meaning has hardly changed from the initial meaning embedded in the root. An index is a means to an end and not the end itself" (Obaseki, 2010).

[6]The positive and negative predictive values are the proportions of positive and negative results in statistics that are true positive and true negative results. Such values describe the performance of a statistical measure.

approaches employ a shallow linguistic analysis[7], based on the use of statistical parsers, in order to analyse users' queries and convert them into a machine-readable format.

In our opinion, the application of a deep linguistic analysis to free-texts and queries represents the possibility to overcome boundaries in KP systems, guaranteeing an improvement of results (di Buono, 2016).

In order to process queries, Halverson *et al.* (2003) propose a mixed approach, involving both tree-based navigation and pattern matching similar to that of structured information retrieval domains.

In his presentation, Lempel[8] (2010) deals with query evaluation strategies, based on Term-at-a-Time (TAAT) and Document-at-a-Time Evaluation (DAAT) processing. TAAT scan postings list one at a time, maintain a set of potential matching documents along with their partial scores. On the other hand, DAAT scan postings lists in parallel, identifying at each point the next potential candidate document and scoring it.

In recent times, and in order to outline concept identification methods, various semantic approaches have been proposed able to assign document ALUs to the correct ontological entries (Sussna, 1993; Baziz *et al.*, 2005; Boubekeur *et al.*, 2010).

Furthermore, different researches employ concept-based systems to process both documents and queries through semantic entities and concepts.

Boubekeur & Azzoug (2013) propose an approach for semantic indexing based on concepts identified starting from a linguistic resource. In their work, the authors use WordNet and WordNetDomains lexical databases, with the aim to identify concepts, and they also apply a concept-based indexing evaluation.

2. System Workflow

In the previous pages, we have presented our LRs, developed by means of LG framework, for the achievement of KP tasks, related to OL and

[7]"The shallow semantic analysis measures only word overlap between text and hypothesis"(Bos & Markert, 2005). This means that starting from tokenization and lemmatization of text and hypothesis, this analysis uses Web documents as corpus and assigns inverse document frequency as weight to each entry in the hypothesis.

[8]From Yahoo! Labs.

population. In order to process unstructured texts and retrieve knowledge, such resources have to be integrated in an NLP environment, therefore being transformed into a lingware.

As previously stated, the AIED and all our other LRs were developed by means of NooJ, a NLP environment which allows to create all four types of Chomsky's grammars. Furthermore, NooJ is suitable to integrate variables in FSA/FSTs, which means that we may generate a complex system based on both LRs and semantic and ontological constraints. In other words, we may integrate variables, in order to achieve a deep linguistic analysis, by means of which we aim at processing knowledge.

After being tested and debugged, the LRs described so far are actually under final development and completion, as part of the NooJ Italian module.

Our methodology relies heavily on a linguistic processing phase, and requires robust resources and background knowledge; it allows the performing of both object/term and synonym identification, and also the recognising of relations. Since it is based on a deep analysis and formalization of linguistic phenomena, our approach can also ensure portability to other knowledge domains, preserving ontology consistency and entity disambiguation.

Therefore, NLP routines, based on Lexicon-Grammar framework, are suitable for supporting the automatic semantic annotation/indexation of textual documents by means of a deep linguistic analysis.

Actually, ESK manages three kinds of inputs, which refer to three independent resources and produce three different outputs. Hereby, we propose its system workflow (Figure 1) which aims at applying a linguistic analysis to three type of inputs, integrating semantic annotation tasks for each one of these:

1. Users' NL query, which refers to LOD KBs, such DBpedia or Europeana repository[9].
2. URL(s), inserted by users, which relates to Web pages and accomplishes a text retrieval.
3. Unstructured texts, uploaded by users, which concern full-text analysis and produce tagged texts.

[9]For more information, see: http://labs.europeana.eu/api/linked-open-data-introduction.

The system workflow is based on representation models applied to all resources, which represent objects of linguistic processing, namely KBs, Web pages and full texts. Therefore, we develop an architecture, which takes advantage from the semantic information stored in LRs and is based on the integration of NooJ. Such system architecture integrates NooJ into a Web application in order to (re)use the representation models outlined in the previous chapter.

Actually, as we have seen, the representation models proposed are developed by means of a semantic annotation process, in order to guarantee the interoperability between structured and unstructured linguistic data. At times, such interoperability is challenging to handle, for the reason that queries may include some restrictions on metadata, such as URL, domain, etc., which may vary from document to document. In order to support these queries, the chosen representation model uses ontological schema to map ALUs with concepts, for avoiding overlapping, and indexing shared content just once. Semantic association is also used to infer Boolean relationship between the elements used in a free-text query, and their specific meta-data.

In other words, starting from the analysis of users' queries and unstructured documents, we employ a semantic annotation process in order to create a match among concepts and their representations.

Furthermore, this architecture may also map linguistic tags (i.e. POS) and structures (i.e. sentences, ALUs) to domain concepts, employing metadata from conceptual schemata. In brief, this means that to achieve such mapping our architecture uses a terminological tagging.

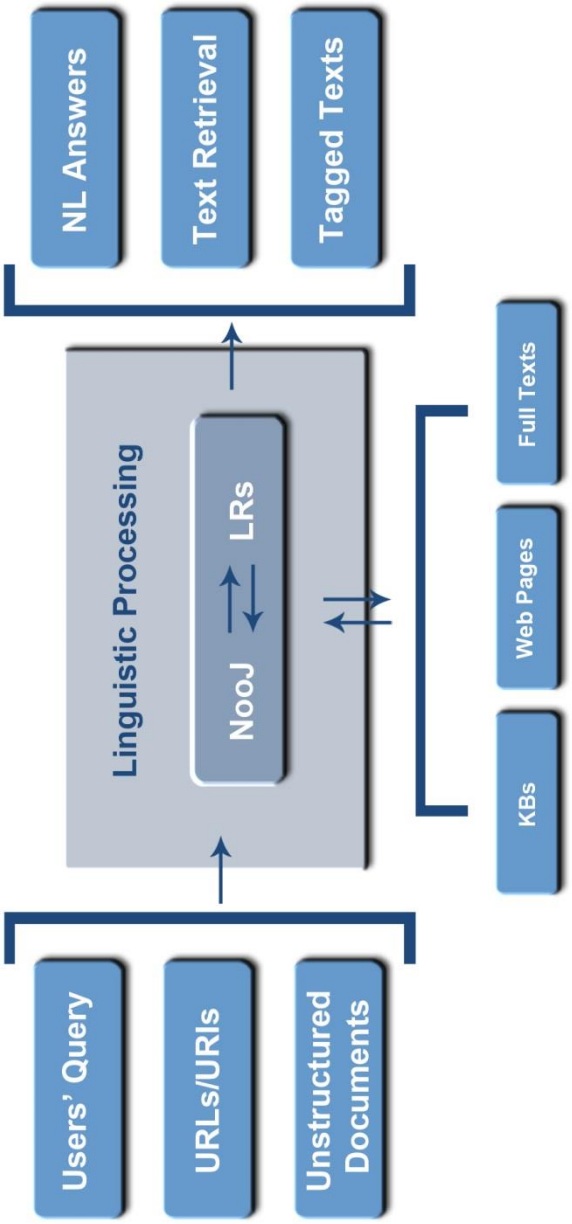


Figure 6.1 - ESK Workflow.

In fact, "terminological tagging represents a central step as regards IR, IE, Machine Translation, ontology development, lexicon-dependent Semantic Web, query-free procedures for knowledge structuring, and also question answering fostering a better «intelligent agent» interaction between humans and technology" (di Buono & Monteleone, 2014). For this reason, representation models, founded on terminological tagging, are suitable to improve existing systems established by the use of both shallow and deep linguistic analysis.

3. System Architecture

ESK[10] (Figure 2) is implemented by dint of PHP code, a server-side scripting language developed for Web application, which is suitable for providing dynamic, user-oriented contents.

ESK is just a beta-version endpoint, built to cope with the Archaeological knowledge domain, and it is set up for Italian. Therefore, the LRs used during the linguistic processing and the source KBs are limited in size and number. This means that, concerning LRs, ESK applies the Italian module of NooJ[11], plus the AIED and a series of related FSA/FSTs. On the other hand, concerning KBs, ESK is structured to run a query against Europeana KBs and DBpedia.

There are several PHP libraries to query SPARQL endpoints; among these, we cite ARC2 library[12], developed by the Knowledge Engineering research laboratory of the Institute for Information Service Science (ISS) within the Center for Computing (CUI) at the University of Geneva.

In our proposal, we integrate PHP code into Web pages in order to manage NooJ Apply[13], so as to provide it with LRs and take inputs from users. In other words, we may run NooJ Apply by means of PHP producing inputs,

[10]http://dsc.unisa.it/mariapiadb/esk/project.html.

[11]For more information, see http://www.nooj-association.org/index.php?option=com_k2&view=item&layout=item&id=54&Itemid=611.

[12]http://cui.unige.ch/isi/icle-wiki/php_sparql_endpoints.

[13]NooJ Apply is a non-commercial version of NooJ, that is a free and open source software, which differs from the version released to the NooJ community, due to the fact that NooJ Apply is a standalone program.

namely LRs and text files, for the accomplishment of the linguistic processing phase.
The system architecture is structured as follows:

1) Users insert input into ESK web form, according to their information needs, which means that they may insert (I) a query, structured as simple sentence; (II) a URL/URI, indicating a Web page to be processed; (III) a text file which they intend to analyse and tag.
2) ESK records inputs into text files and applies a normalization process, i.e., it deletes punctuation marks and so on.
3) ESK handles such text files as inputs for NooJ Apply, and it also manages LRs which have to be associated to the running linguistic analysis.
4) NooJ Apply performs linguistic processing, which is different for each of the three inputs. As we will see, this means that ESK applies different LRs, namely FSA/FSTs, in order to return different outputs.
5) If the input is a NL question, after the linguistic processing phase, ESK runs the corresponding query against a remote endpoint proceeding as it follows:

    a. Open and access to the endpoint;
    b. Execute the query;
    c. Display the results.

6) ESK may return the following different results:

    a. An NL answer to users' query. This means a list of results as literals, which cope with the request and references of resources for these results.
    b. A Text Retrieval, which is intended as (a set of) semantically annotated strings, referring to ALUs and concepts, extracted from the page indicated by users' URL.
    c. A Tagged Text. This means a document annotated in XML and also by metadata schemata, as they were introduced in Chapter I, e.g., RDF, EDM and SKOS.

d. Additionally, ESK may also handle the results of the statistical analysis performed by means of NooJ Apply, e.g., frequencies, standard score and so on[14].

In the following pages, we will present the linguistic processing phase for (and the LRs applied to) each inputs, together with the corresponding results.

[14]It is worth noting that NooJ allows to analyse also selected parts of corpora, thus we may apply statistical measures to only these parts, so obtaining focused measures.

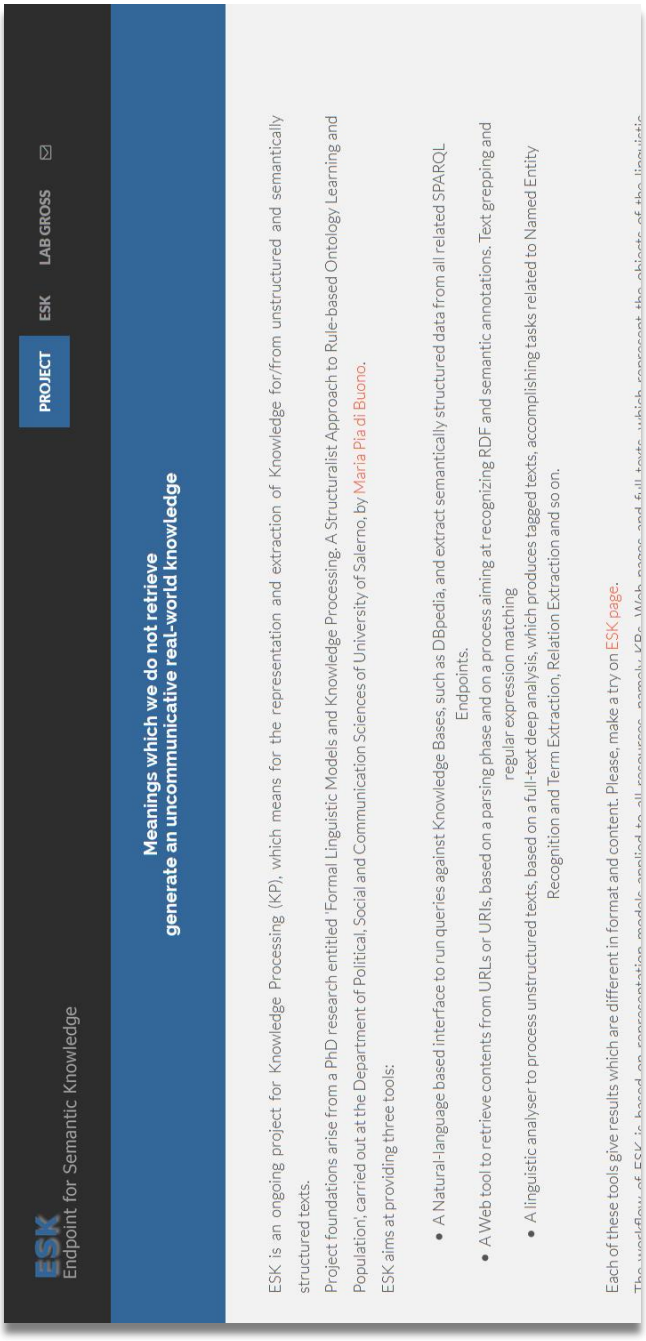


Figure 6.2 - ESK Homepage.

4. The Linguistic Processing

Users' Query

In order to process this fist type of input, the main task to accomplish is the processing of user's query by means of a linguistic analysis. Such an analysis aims at annotating the query using a domain-independent semantic data model (i.e., DBpedia cross-domain ontology).

Thus, ESK records the query on a text file and set it as a NooJ parameter, namely as a variable used as one of the inputs to the subroutine.

During this analysis, other variables are assigned to NooJ subroutine, namely the AIED and a set of grammars in the form of FSA.

In Figure 3, we present a sample of FSA/FST which may be used to process a NL query. In such an example, the query is devoted to extract the name and surname of a person who makes a specific profession and is living during a specific period.

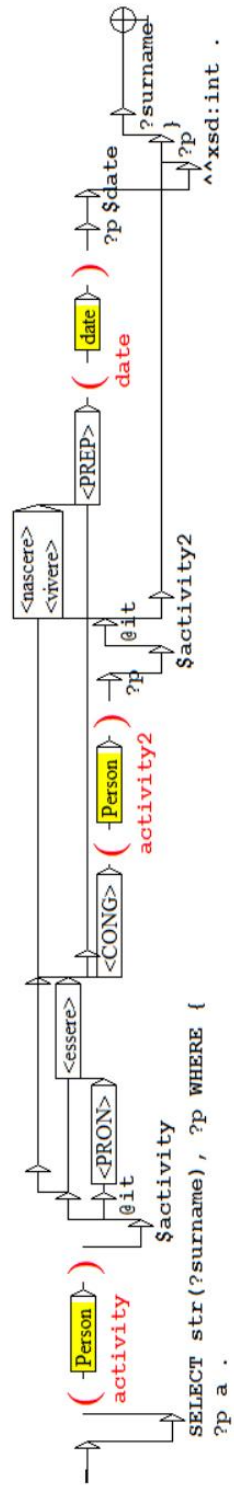


Figure 6.3 - FSA for annotating users' queries.

In other words, the previous automaton may process a query as the following one:

(1) *Tutti gli archeologi che sono stati anche scrittori nati nel '900*[15] (All the archaeologists who have been also writers and were born in 19th century).

Such a kind of query represents a reduction of two simple sentences, which are:

(1a) N_{um} *che svolgono l'attività di archeologo e di scrittore*
(Persons who are archaeologists and writers).
(1b) N_{um} *che sono nati nel '900*
(Persons who were born in the 19th century).

Both these two sentences may be structured and formalized on the basis of the First-Order Logic:

(1c) To be (x,y) ⊃ Person (x)
To be (x,y) ⊃ Activity (y)
(1d) To live (x,y) ⊃ Person (x)
To live (x,y) ⊃ Date (y)

Applying an Equi-NP deletion[16] to (1a) and (1b), we may delete the NP of the second complement clause if it is co-referential with the subject or object of the main clause. In other words, in (1) we delete one of the two identical NPs (N_{um}/Person), which means that in (1b) and (1d) we may delete the NP which is co-referential with (1a) and (1c).

Such a formalization refers to an independent-domain model, suitable to retrieve information from a blanket RDF-based KB, for instance BDpedia.

[15]Such example is adapted from the one proposed on the page of Italian DBpedia.

http://it.dbpedia.org/esempi/.

[16]Equi-Np Deletion is a rule of transformational grammar that deletes the subject of a coordinate or complement clause if it is coreferential with the subject or object of the main clause, or of another coordinate clause.

Therefore, the automaton in Figure 3 allows us to recognize entities involved in RDF relationships and occurring in (or recalled by) the query context, respectively *Person* and *Activity* and *Person* and *Date*.

In such RDF triples, the subject, *Person*, and the objects, *Activity* and *Date*, are trigged by two predicate, namely two VPs. As we have seen, these VPs are represented by two verb classes, i.e., *to be* and *to live* or (E + *to be*) *born*, which may co-occur together with the given entities in sentence contexts.

It is worth noticing that in our sample we insert two nodes containing the same entity (*Person*), which stands for two different variables, namely *Activity* and *Activity2*. Such variables refer to a specific tag, which is used to identify a specific attribute, namely a profession, for the elements belonging to the generic class *Person*.

Values, produced by variables (*Activity*, *Activity2* and *Date*) are employed to generate a SPARQL query. Such a query is suitable to retrieve the name and surname of such persons who are performing a specific activity/job/profession during a determinate time lapse.

Therefore, the output of the FSA may be used in order to generate a query which may be run against any SPARQL endpoint or repository in which documents are formalized using RDF graphs.

The following sample shows the result of the FSA when applied to the previous query:

```
SELECT ?name, ?surname
WHERE {
      ?p a .
      ?p “scrittore”@it .
      ?p “archeologo”@it .
      ?p “1900”^^xsd:int .
      ?p ?surname
}
```

[Example of pseudo-code query in SPARQL which may be used against an RDF KBs]

The previous FSA output indicates how, in the graph of DBpedia KBs, we find all subjects (Person) and objects (Activity/Activity2 and Date) linked with the *to be* and *to live* predicate. The returned values have to correspond to all

the values of ?name and ?surname. In other words, the resulting output means *find all surnames and names of all the persons who were archaeologists and writers and lived during the 19th century*.

In Paragraph 5, we analyse how such result may be used in a SPARQL architecture when we run the given query against a KBs.

URLs/URIs

In presence of a specific user's query, ESK connects to a Web page and save its contents into a text file, applies to it a normalization process and consequently set it as a NooJ variable. During the linguistic processing of such kind of inputs, we apply the AIED to recognize ALUs/entities and associate (L)LOD references to each one of them.

Actually, this second subtask, namely data representation, involves appropriate operations on the RDF-based data layer, which includes the mapping of OWL concepts to object-oriented classes with methods for interrelations and domain-specific rules, used to generate and consolidate all processes (e.g. CIDOC CRM ontology).

Such process of data representation aims at analysing the information stored in Web documents, which means that we may directly retrieve information from any URLs.

According to our FSA-based approach, electronic dictionaries entries (simple words and ALUs) are the subject and the object of the RDF triples which are traceable inside sentence structures.

In addition, as regards declarative sentences, RDF gives the possibility to recognize sentences conveying information of the type "X is an element of Y", which also is a type of recursive and iterative (therefore productive) structure.

Actually, inside FSA, we enclose and use RDF data representation and CIDOC CRM. Such an enclosure is presented in Chapter V, by means of which we identify and extract entities and properties.

In other words, we retrieve and extract ALUs/entities and VPs/properties from texts using a formalization based on a match among elements in nuclear sentence and the domain-specific ontology.

In fact, if we formalize the previous query sample (1) by means of the CIDOC CRM, we obtain:

(1d) E21(Person)+P14(*perform*)+E7(Activity1/2)

(1e) E21(Person)+P98(*was born*)+E50(Date).

Therefore, ESK results for a URL-based input is represented by a list of elements, namely entities, retrieved from the queried Web page, and structured as follows:

<E21><Person>Peter Chad Tigar Levi**</Person></E21>** (…)
<P14></P14><E50><Date>16 May 1931 – 1 February 2000**</Date></E50>**
<P98>was**</P98>** (…)
<E7><Activity1>archaeologist**</Activity1>** (…)
<Activity2>travel writer**</Activity2></E7>**[17].

Therefore, starting from the entries retrieved and their specific information, stored in the AIED and in the FSA/FSTs, we label entities and properties directly using the CIDOC CRM schema. This procedure automatically generates the tags, while correctly coupling ontology constituents and elements in the text.

Furthermore, considering that users may insert also a URI in the input field, we develop an FSA suitable to retrieve such kind of information (Figure 4). In the FSA, we use the nodes on the left in order to recognize labels used inside RDF documents, which are stored, for example, in DBpedia KB. This means that:

- First, we process tags which describe elements semantically.
- Subsequently, we analyse which values are assumed for such descriptions (literal or numeric).

Actually, the FSA presents two paths: literal or numeric. For literal paths, in the subsequent node, we insert a generic <WF> class, namely a blanket Word Form, in order to recognize each word form which is present inside documents. These word forms represent values which are stored for each specific semantic descriptive tag, i.e. for the *foaf:surname* Levi value (di Buono, 2016).

[17]https://en.wikipedia.org/wiki/Peter_Levi.

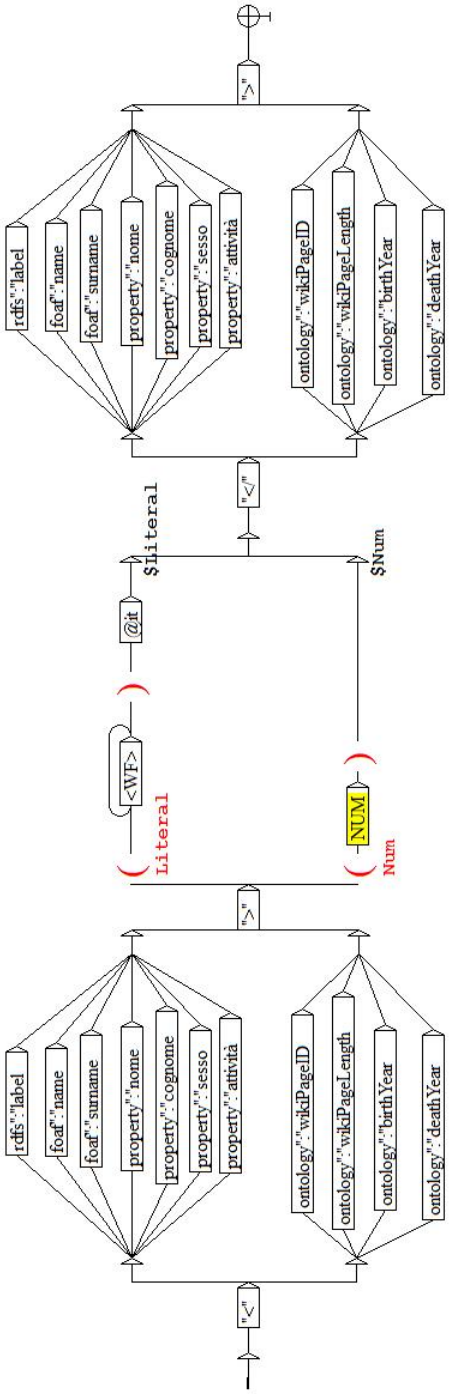


Figure 6.4 - Sample of FSA for structured-text analysis.

On the other hand, the final node *@it* indicates the language tag in the resource description schemata (i.e., Italian). Actually, it is worth noticing that DBpedia routinely provides the language attribute for literals. Therefore, a query containing a literal must have an "@" language tag.

In the FSA, the second path is suitable to recognize numeric values by means of a recursive sub-graph without language tag.

For this kind of input, ESK returns result in the form of structured information stored in the URI used. Thus, we may retrieve information within a URI document, as shown in the following sample:

Table 6.1 - Sample of results from URI-content processing.

Path	**Output**
<rdfs:label> Peter Levi @it</rdfs:label>	Peter Levi @it
<ontology:wikiPageID>2168662</ontology:wikiPageID>	2168662
<foaf:name> Peter Chad Tigar @it </foaf:name>	Peter Chad Tigar @it
<ontology:wikiPageLength>11646</ontology:wikiPageLength>	11646
<ontology:birthYear>1931</ontology:birthYear>	1931
<ontology:deathYear>2000</ontology:deathYear>	2000
<foaf:surname> Levi @it </foaf:surname>	Levi @it
<property:nome> Peter Chad Tigar @it </property:nome>	Peter Chad Tigar @it
<property:cognomen> Levi @it </property:cognomen>	Levi @it
<property:sesso> M @it </property:sesso>	M @it
<property:attività> Scrittore @it</property:attività> <property:attività> … </property:attività>	Scrittore @it … Poeta @it … Archeologo @it

Unstructured Texts

ESK also offers a function suitable for processing a text file uploaded by users, applying Archaeological LRs. Actually, during this phase, using FSA as input variables, NooJ allows to convert an unstructured text into a document formalized according to a specific-domain data model.

Furthermore, ESK guarantees the possibility to export such results, using RDF and SKOS, and also to use LLOD URIs to tag the AIED entries. As we have seen in Chapter IV, these characteristics are two of the main features by means of which our system for ontology semi-automatic population is built (di Buono *et al*., 2014b).

This procedure for unstructured-text processing is accomplished by means of the following steps:

1. NooJ processes a text, parses it, and locates all the terminological ALUs inside the given text;
2. If retrieved ALUs belong to *Agent* or *Place* classes, we associate them to URIs in order to integrate LLOD resources (see Chapter V).
3. Subsequently, the ALUs retrieved are conceptually described by means of SKOS/RDF schemata and features, as for instance those used in EDM;
4. At the same time, RDF triples are transformed into EDM tags in which concepts and relationships, for instance E21 or P14, are rewritten by means of corresponding “edm:Agent” or “edm:begin”;
5. Finally, NooJ output is transformed into a full EDM XML Schema.

In Figure 5, we present an FSA which allows to convert unstructured text into a data model based on EDM XML Schema. Such an FSA recognizes NPs and VPs, associating them to CIDOC CRM classes and property and, subsequently, labelling these elements with tags of the EDM schema.

For instance, by means of this FSA, we may recognize sentences such as the one in the sample (1b). Therefore, ESK converts this unstructured text into a string structured according to EDM XML Schema, as in the result which follows:

```
<edm:Agent>Num</edm:Agent>
<edm:begin>born in</edm:begin>
```

```
<edm:TimeSpan>Date</edm:TimeSpan>
```
[18].

It is worth to remember that such a result comes from the ontological and semantic constraints which are applied during the formalization process. In other words, the VP requires the co-occurrence of an N_{um} as N_0 and a Date as N_1, due to the semantic behaviour assumed by the verb.

Furthermore, our procedure for (L)LOD integration (Chapter V) is also directed to insert (L)LOD references into the output deriving them from unstructured-text processing. In other words, when an ALU is recognized as an element belonging to a CIDOC CRM class, such as Agent/Person or Place, ESK appends the corresponding URI, automatically generated.

In the example previously given, if the NP stands for an N_{um}, ESK returns the following result:

```
<edm:Agent
rdf:about="http://it.dbpedia.org/resource/Peter_Levi
">
Peter Levi
</edm:Agent>
```

To sum up, by means of NooJ FSA, we create a matching among different formal models, suitable to generate structured textual data and, at the same time, associate them to (L)LOD.

[18]It is worth noticing that in such example we do not use *edm:year* in order to tag birth/death date, due to the fact that such property refers to an event in the life of the original analogue or born digital object. Therefore, *edm:year* property is not applicable to the class *Agent*.

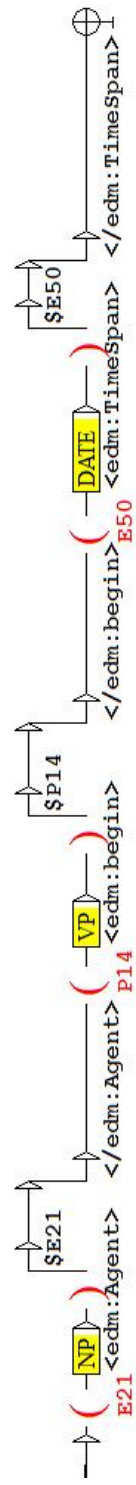


Figure 6.5 – Sample of FSA for unstructured-text processing.

5. SPARQL Architecture & Endpoints

As we stated previously, the result of the linguistic processing applied to a given user's query is suitable for generating a SPARQL query which may be run against a KB.

Generally speaking, SPARQL queries are executed against RDF datasets, consisting of RDF graphs. Such queries are accepted by a SPARQL endpoint which returns results via HTTP. There exist two kinds of SPARQL endpoints: generic, which query any Web-accessible RDF data; and specific, which are hardwired to query against particular datasets.

A SPARQL query has to be structured in the way which follows (strings which start with # stand for comments into code):

```
# prefix declarations
PREFIX foo: <http://example.com/resources/>
...
# dataset definition
FROM ...
# result clause
SELECT ...
# query pattern
WHERE {
    ...
}
# query modifiers
ORDER BY ...
```

[Example of pseudo-code query in SPARQL which may be used into an Endpoint]

- In order to run a query the *PREFIX*, declarations are mandatory, due to the fact that they indicate the KB(s) in which RDF graphs are stored. Therefore, results from the linguistic processing are further processed, in order to insert such declarations before the SELECT statement, as for instance it happens with the following prefixes:

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
PREFIX dbres: <http://dbpedia.org/resource/>
```

- SPARQL *variables* start with a ? and can match any node (resource descriptors, such as a URI or a URL, or literal values) in the RDF dataset.
- *Triple patterns* are just like triples, except that any of the parts of a triple can be replaced with a variable.
- The *SELECT* result clause returns a table of variables and values that satisfy the query. In other words, the SELECT query form is used to create a list of URIs which satisfy the pattern-matching requirements specified in the query.

Usually, the results of SPARQL queries can be returned and/or rendered in a variety of formats:

- XML. SPARQL specifies an XML vocabulary for returning tables of results.
- JSON. A JSON "port" of the XML vocabulary, particularly useful for Web applications.
- RDF. Certain SPARQL result clauses trigger RDF responses, which in turn can be serialized in a number of ways (RDF/XML, N-Triples, Turtle, etc.)
- HTML. When using an interactive form to work with SPARQL queries. Often implemented by applying an XSL transform to XML results.

After processing the query, ESK opens the connection to the KB and runs the query, namely the result of linguistic processing. Consequently, ESK displays the results as a table which contains name and surname values, i.e. Peter Levi, and the specific resource URL for such values.

Thus, for the given query we obtain a list of literals and the corresponding RDF pages which match with the user's information needs.

Table 6.2 - Results from SPARQL Query.

Name/Surname Value	Resource
Peter Levi	http://it.dbpedia.org/resource/Peter_Levi
Paolo Matthiae	http://it.dbpedia.org/resource/Paolo_Mattiae
Thorkild Hansen	http://it.dbpedia.org/resource/Thorkild_Hansen
Glenn Cooper	http://it.dbpedia.org/resource/Glenn_Cooper
Alfred Duggan	http://it.dbpedia.org/resource/Alfred_Duggan
Max Mallowan	http://it.dbpedia.org/resource/Max_Mallowan
Almerico Meomartini	http://it.dbpedia.org/resource/Almerico_Meomartini
Michael Coe	http://it.dbpedia.org/resource/Michael_D._Coe
Thanos Kondylis	http://it.dbpedia.org/resource/Thanos_Kondylis
Vincenzo Zecca	http://it.dbpedia.org/resource/Vincenzo_Zecca
En Bellis	http://it.dbpedia.org/resource/En_Bellis
Sebastiano Consoli	http://it.dbpedia.org/resource/Sebastiano_Consoli

6. Tests and Evaluation

In our experiment, we use DBpedia database as a knowledge source of structured data in RDF/XML, and we test our system outputs using its SPARQL (Protocol and RDF Query Language) Endpoint.
There also is a public SPARQL endpoint over the DBpedia data set and, as reported in the site, users can run queries against DBpedia using:

- The Leipzig query builder;
- The OpenLink Interactive SPARQL Query Builder (iSPARQL);
- The SNORQL query explorer; or
- Any other SPARQL-aware client(s).

Therefore, DBpedia endpoints may be accessed just using a query encoded in SPARQL. We tested our system outputs, i.e., SPARQL queries and data representations, using Italian DBpedia KB.

We also test our data representation, obtained through NooJ FSA, on a corpus dumped from the Italian Wikipedia Database.

We evaluated each kinds of results produced by ESK, providing for the three outputs Precision, Recall and F-Score values.

Table 6.3 – ESK Evaluation.

Output type	Precision	Recall	F-score
NL Answers	0.75	0.52	0.61
Text Retrieval	0.83	0.51	0.63
Tagged Texts	0.96	0.51	0.67

Such measures are suitable for evaluating the validity of our method during the accomplishment of each input. At present, we still do not produce a full analysis for error sources which cause a decrease of ESK performance. Generally speaking, some challenging aspects in the linguistic processing phase concern discourse analysis[19]. This means that ESK returns significant results if applied to nuclear sentences, within both users' query and text processing. The performance concerning text retrieval is evaluated without distinguishing URL results from URI ones, because we choose to estimate the total performance related to the input kind.

As we can notice, the values present a variability with reference to the different outputs. Anyway, we consider the tagged texts results very satisfying; however, we are already planning to enrich our research outcomes with several additional improvements.

All the annotations produced by the application of our method and resources can be reused to enrich lexical databases or ontologies referred to the CH domain. Noticeably, the size and quality of the enrichment is strictly dependent on the largeness and on the content of the corpus on which the NooJ resources are applied. Therefore, in order to obtain widespread CH

[19]For the definition of discourse see Harris (1952 and 1970).

databases, it is preferable to use corpora able to cover the larger group of CH domain possible.

Our future research work aims at integrating different RDF formats in the parser and writer registries, i.e. Turtle, JSON-LD, RDF/JSON and so on.

Future work also aims at integrating manually constructed rules with supplementary rules, in order to improve not-probable word removal. In addition, we are planning to develop grammars useful to recognize discontinuous expressions inside NPs and VPs, and to implement an anaphora-resolution task.

CONCLUSIONS AND FUTURE WORKS

At the conclusion of this dissertation, we want to summarise the work presented in the previous chapters, also describing the future directions our research will take.

After many years of research and improvements together with the adoption of different approaches in KR and KE, ontology learning and population still represent a critical area in current technologies. Due to the intrinsic semantic properties specific to knowledge, ontology learning and population may give rise to many inaccuracies, which can seriously reduce the precision and quality of KP outputs.

In this dissertation, we have tried to show that a structuralist approach to KP, by means of a precise analysis and comparison of natural languages and machine formalisms, can improve the processing task as far as ontology learning and population are concerned.

In fact, significant improvements in KP quality have been achieved since the introduction of ontologies; anyway, knowledge treatment still presents important shortcomings. If KP intends to achieve the development of concretely useful tools, in both representation and extraction tasks, it has to tackle the problems posed by human/machine-language formalization and provide an adequate processing approach to such formalisms. If it does not, it will fail to produce high quality outputs.

This work has presented the ongoing theoretical discussion concerning different aspects of KP, such as its definition, representation and extraction models and methods, illustrating different approaches to the issue, i.e. stochastic and rule-based methods.

Therefore, based on the Lexicon-Grammar theoretical framework, our experiment provides, on the one hand, an investigation of a broad variety of KP methods and, on the other hand, a representation methodology that foresees the interaction of LRs and machine formalisms to efficiently handle knowledge.

We proposed a model of representation, retrieval and extraction of knowledge, based on the assumption that it is necessary to use a formal

semantics description, converting it into a machine-readable formal representation.

In this model, we may identify ontologies as the trigger between Human World and Machine World representations. Indeed, due to the fact that they focus on terms meaning and on the nature and structure of a given domain, ontologies are suitable to match human and machine representations.

Therefore, during KP, that is during the steps of representation, retrieval and extraction, we always have to keep in mind our attempt to create a trigger between Human World and Machine World.

This research work has then produced two main results in the field of KP so far. First, it has led to the development of both AIED first version and other LRs, as thoroughly described in Chapter V of this work. Second, it has led to the development of ESK, a beta version of a Web endpoint suitable to both formalize and extract knowledge from (semi-)structured and unstructured texts.

The whole work is based on an iterative and extendable method based on LRs that allow a deep semantic analysis, the core of both a prototype question-answering system and an on-line tool for ontology learning and population.

A fine-grained linguistic analysis, achieved by means of NooJ, has a crucial role in developing effective processing methodologies that enable a precise and meaningful KP.

For our future work, we plan to further investigate formalization of natural languages from a LG perspective, particularly with respect to domain-specific linguistic features and machine-language equivalences.

Thus, we aim at improving both an index-data structuring and a query evaluation process. It is also necessary to test the system in a consistent way, on other KBs, in order to propose an independent-domain approach.

Our long-term goal is to integrate our method with a hybrid approach to KP, in order to achieve high quality knowledge representation and extraction by combining probabilistic and linguistic information.

However, to achieve this goal, we must devise additional efficient strategies for representing deep attributes and semantic properties of natural languages also concerning machine formalisms.

Furthermore, we must consider both theoretical and practical aspects of the computational treatment of knowledge focusing on new applicative settings of ESK.

In conclusion, the focus of this research for the coming years will be to improve the results obtained so far and to extend the research work providing a more comprehensive methodology for ontology learning and population in KP.

Anyway, it is unmistakable that not all natural languages can be formalized in their entirety due to fact that they are not static and finite sets. Therefore, even if not all linguistic production may be managed by means of machines, independently from the formalization approach encompassed, we firmly believe that a comprehensive and analytic formalization may significantly improve current KP methodologies.

REFERENCES

Abeillé, A. (1988). Parsing French with tree adjoining grammar: some linguistic accounts. In *Proceedings of the 12th conference on Computational linguistics-Volume 1* (pp. 7-12). Association for Computational Linguistics.

Abney, S. P. (1992). *Parsing by chunks* (pp. 257-278). Springer Netherlands.

Acar, E., & Yener, B. (2009). Unsupervised multiway data analysis: A literature survey. *Knowledge and Data Engineering, IEEE Transactions on*, *21*(1), 6-20.

Ackoff, R. L. (1989). From data to wisdom. *Journal of applied systems analysis*, *16*, 3-9.

Agrawal, R., & Srikant, R. (2001). On integrating catalogs. In *Proceedings of the 10th international conference on World Wide Web* (pp. 603-612). ACM.

Ahmad, A., & Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, *63*(2), 503-527.

Ajdukiewicz, K. (1935). Die syntaktische konnexiẗat. In Storrs McCall, ed., *Polish Logic* 1920-1939, 207–231. Oxford: Oxford University Press. Translated from Studia Philosophica, 1, 1-27.

Al Hasan M. (2014) Discrete Sequence Classification in Aggarwal C.C. (ed.) 2014 *Data Classification: Algorithms and Applications*. CRC Press.

Aldenderfer, M. S., & Blashfield, R. K. (1984). Cluster Analysis. Sage University Paper Series On Quantitative Applications in the Social Sciences 07-044.

Aldini A. (2014) Formal languages and software verification, Lectures DiSBeF – Sezione STI, University of Urbino "Carlo Bo" – Italy http://www.sti.uniurb.it/aldini/publications/flsv.pdf

Anderberg, M. R. (2014). *Cluster Analysis for Applications: Probability and Mathematical Statistics: A Series of Monographs and Textbooks* (Vol. 19). Academic press.

Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., & Tyson, M. (1993). FASTUS: A finite-state processor for information extraction from real-world text. In *IJCAI* (Vol. 93, pp. 1172-1178).

Aussenac-Gilles, N., Despres, S., & Szulman, S. (2008). The terminae method and platform for ontology engineering from texts. *Bridging the Gap between Text and Knowledge-Selected Contributions to Ontology Learning and Population from Text*, 199-223.

Äyrämö, S., & Kärkkäinen, T. (2006). Introduction to partitioning-based clustering methods with a robust example.

Bach, E. (1976). An extension of classical transformational grammar. In *Problems in Linguistic Metatheory* (Proceedings of the 1976 conference). East Lansing, Michigan: Michigan State University, 183–224.

Bachimont, B. (2000). Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des connaissances: évolutions récentes et nouveaux défis*, 305-323.

Bagheri Hariri, B., Abolhassani, H., & Sayyadi, H. (2006). A neural-networks-based approach for ontology alignment. In *SCIS & ISIS* (Vol. 2006, No. 0, pp. 1248-1252). 日本知能情報ファジィ学会.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1* (pp. 86-90). Association for Computational Linguistics.

Baldridge, J. (2002). *Lexically specified derivational control in combinatory categorial grammar* (Doctoral dissertation, University of Edinburgh).

Banfield, J. D., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 803-821.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction for the web. In *IJCAI* (Vol. 7, pp. 2670-2676).

Banko, M., Etzioni, O., & Center, T. (2008). The Tradeoffs Between Open and Traditional Relation Extraction. In *ACL* (Vol. 8, pp. 28-36).

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. Language, 29, 47–58.

Baziz, M., Boughanem, M., & Aussenac-Gilles, N. (2005). A conceptual indexing approach based on document content representation. Dans les actes de COLIS 2005 Context: nature, impact and role, Glasgow, Grande-Bretagne, 171–186. LNCS 3507.

Bellinger, G., Castro, D., & Mills, A. (2004). Data, information, knowledge, and wisdom.

Bengfort, B. (2013). An Introduction to Named Entity Recognition in Natural Language Processing. White paper. http://www.datacommunitydc.org/blog/2013/04/a-survey-of-stochastic-and-gazetteer-based-approaches-for-named-entity-recognition.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137-1155.

Bergenholtz, H., & Tarp, S. (Eds.). (1995). *Manual of specialised lexicography: the preparation of specialised dictionaries* (Vol. 12). John Benjamins Publishing.

Berners-Lee, T. (2006). Linked Data – Design Issuses. *W3C*. www.w3.org/DesignIssues/LinkedData.html

Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., ... & Sheets, D. (2006). Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop* (Vol. 2006).

Bezdek, J. C. (2013). *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media.

Bian, J., Gao, B., & Liu, T. Y. (2014). Knowledge-powered deep learning for word embedding. In *Machine Learning and Knowledge Discovery in Databases* (pp. 132-148). Springer Berlin Heidelberg.

Biemann, C. (2005). Ontology Learning from Text: A Survey of Methods. In *LDV forum* (Vol. 20, No. 2, pp. 75-93).

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205-227.

Bordoni, L., Costantini, M., & Curcio, A. (2013). A case study in archaeological documentation with ontological modeling. In *Conference on Cultural Heritage and New Technologies, Vienna*.

Bos, J., & Markert, K. (2005). Combining shallow and deep NLP methods for recognizing textual entailment. In *Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK* (pp. 65-68).

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). ACM.

Boubekeur, F., & Azzoug, W. (2013). Concept-based indexing in text information retrieval. *arXiv preprint arXiv:1303.1703*.

Boubekeur, F., Boughanem, M., Tamine, L., & Daoud, M. (2010). Using WordNet for Concept-based document indexing in information retrieval. In *Fourth International Conference on Semantic Processing (SEMAPRO), Florence, Italy*.

Brachman, R., & Levesque, H. (2004). *Knowledge representation and reasoning*. Elsevier.

Brame, M. (1981). Trace theory with filters vs. lexically based syntax without. *Linguistic Inquiry*, 275-293.

Brandes, U., Eiglsperger, M., Lerner, J., & Pich, C. (2010). *Graph markup language (GraphML)*. Bibliothek der Universität Konstanz.

Breiman, L. (1996). Bias, variance, and arcing classifiers (Technical Report 460). *Statistics Department, University of California*.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Bresnan, J. (1982). *The mental representation of grammatical relations* (Vol. 1). The MIT Press.

Brewster, C., & O'Hara, K. (2007). Knowledge representation with ontologies: Present challenges—Future possibilities. *International Journal of Human-Computer Studies*, *65*(7), 563-568.

Briscoe, T., Carroll, J., & Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL on Interactive presentation sessions* (pp. 77-80). Association for Computational Linguistics.

Budanitsky, A., & Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, *32*(1), 13-47.

Bühmann, L., & Lehmann, J. (2012). Universal OWL axiom enrichment for large knowledge bases. In *Knowledge Engineering and Knowledge Management* (pp. 57-71). Springer Berlin Heidelberg.

Bühmann, L., & Lehmann, J. (2013). Pattern based knowledge base enrichment. In *The Semantic Web–ISWC 2013* (pp. 33-48). Springer Berlin Heidelberg.

Buitelaar, P., Choi, K. S. K., Cimiano, P., & Hovy, E. (2012). The Multilingual Semantic Web. Technical Report 12362, Report from the Dagstuhl Seminar.

Buitelaar, P., Olejnik, D., & Sintek, M. (2004). A protégé plug-in for ontology extraction from text based on linguistic analysis. In *The Semantic Web: Research and Applications* (pp. 31-44). Springer Berlin Heidelberg.

Caliusco, M. L., & Stegmayer, G. (2010). Semantic web technologies and Artificial Neural Networks for intelligent web knowledge source discovery. In *Emergent Web Intelligence: Advanced Semantic Technologies* (pp. 17-36). Springer London.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E. R., & Mitchell, T. M. (2010). Toward an Architecture for Never-Ending Language Learning. In *AAAI* (Vol. 5, p. 3).

Carnie, A. (2013). *Syntax: A generative introduction*. John Wiley & Sons.

Chakraborty, A.R., & Chakrabarti, B. (1983). INDEXING: Principles, process, and products . Calcutta: World Press Private Limited.

Chambers, N., & Jurafsky, D. (2011). Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of*

the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 976-986). Association for Computational Linguistics.

Chiarcos, C., Declerck, T., & Mccrae, J. P. (2013). Linguistic Linked Open Data (LLOD) Introduction and Overview. *Proceedings of LDL 2013.*

Chiarcos, C., Hellmann, S., & Nordhoff, S. (2012). Linking linguistic resources: Examples from the open linguistics working group. In *Linked Data in Linguistics* (pp. 201-216). Springer Berlin Heidelberg.

Chiarcos, C., McCrae, J., Cimiano, P., & Fellbaum, C. (2013b). Towards open data for linguistics: Linguistic linked data. In *New Trends of Research in Ontologies and Lexical Resources* (pp. 7-25). Springer Berlin Heidelberg.

Chieu, H. L., & Ng, H. T. (2002). Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (pp. 1-7). Association for Computational Linguistics.

Chilton, L. B., Little, G., Edge, D., Weld, D. S., & Landay, J. A. (2013). Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1999-2008). ACM.

Chinchor, N., & Robinson, P. (1997). MUC-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding* (p. 29).

Chiticariu, L., Li, Y., & Reiss, F. R. (2013). Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!. In *EMNLP* (pp. 827-832).

Chomsky, N., (1957). *Syntactic Structures.* The Hague: Mouton.

Chomsky, N., (1965). Aspects of the theory of syntax Cambridge. *Multilingual Matters: MIT Press*.

Chomsky, N. (1981). *Lectures on government and binding: The Pisa lectures* Holland: Foris Publications. Reprint. 7th Edition. Berlin and New York: Mouton de Gruyter, 1993.

Chomsky, N. (1986). *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group.

Chomsky, N. (1995). *The minimalist program* (Vol. 1765). Cambridge, MA: MIT press.

Church, K. W. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing* (pp. 136-143). Association for Computational Linguistics.

Cimiano, P. (2006). *Ontology learning from text* (pp. 19-34). Springer US.

Cimiano, P., & Völker, J. (2005). Text2Onto. In *Natural language processing and information systems* (pp. 227-238). Springer Berlin Heidelberg.

Cimiano, P., Mädche, A., Staab, S., & Völker, J. (2009). Ontology learning. In *Handbook on ontologies* (pp. 245-267). Springer Berlin Heidelberg.

Cireşan, D. C., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013* (pp. 411-418). Springer Berlin Heidelberg.

Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, *22*(12), 3207-3220.

Clerkin, P., Cunningham, P., & Hayes, C. (2002). *Ontology discovery for the semantic web using hierarchical clustering*. Trinity College Dublin, Department of Computer Science.

Coburn, E., Light, R., McKenna, G., Stein, R., & Vitzthum, A. (2010). Lido-lightweight information describing objects version 1.0. *ICOM International Committee of Museums*.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167). ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, *12*, 2493-2537.

Cowie, J., & Lehnert, W. (1996). Information extraction. *Communications of the ACM*, *39*(1), 80-91.

Cox, D. R. (1958). Some problems connected with statistical inference. *The Annals of Mathematical Statistics*, 357-372.

Cox, D. R. (1958b). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 215-242.

Culotta, A., McCallum, A., & Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (pp. 296-303). Association for Computational Linguistics.

Cunningham, H. (1999). A definition and short history of Language Engineering. *Natural Language Engineering*, *5*(1), 1-16.

Cunningham, H. (2005). Information extraction, automatic. *Encyclopedia of language and linguistics,*, 665-677.

Curry, H. B., & Feys, R. (1958). Studies in Logic and the Foundations of Mathematics. In *Combinatory logic* (Vol. 1). North-Holland Amsterdam.

D'Agostino E. & Elia A. (1998). Il significato delle frasi: un continuum dalle frasi semplici alle forme poliremathiche. In Albano Leoni *et al.* (eds.) *Ai limiti del linguaggio.* Roma/Bari:Laterza.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, *20*(1), 30-42.

Daniel, J., & James, H. M. (2000). Speech and Language processing: An introduction to natural language processing. *Computational Linguistics and Speech Recognition. Prentice Hall, NJ, USA*.

Das, S. K., Kumar, A., Das, B., & Burnwal, A. P. (2013). ON SOFT COMPUTING TECHNIQUES IN VARIOUS AREAS. *Computer Science & Information Technology*, 59.

Davenport, T. H., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Harvard Business Press.

Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a knowledge representation?. *AI magazine*, *14*(1), 17.

Debusmann, R. (2000). An introduction to dependency grammar. *Hausarbeit fur das Hauptseminar Dependenzgrammatik SoSe*, *99*, 1-16.

Debusmann, R., Duchier, D., & Kruijff, G. J. M. (2004). Extensible dependency grammar: A new methodology. In *Proceedings of the COLING 2004 Workshop on Recent Advances in Dependency Grammar* (pp. 70-76).

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JAsIs*, *41*(6), 391-407.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1-38.

Denber, M. (1998). Automatic resolution of anaphora in English. *Eastman Kodak Co*.

Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine learning*, *42*(1-2), 143-175.

di Buono, M.P. (2015). Information Extraction for Ontology Population Tasks. An Application to the Italian Archaeological Domain. *International Journal of Computer Science: Theories and Applications*. Vol 3, No 2. ORB Academic Publisher.

di Buono, M.P. (2016) Semi-Automatic Indexing and Parsing Information on the Web with NooJ. Selected papers in Proceedings of NooJ 2015 International Conference 2015. Springer

di Buono, M. P., Monteleone, M., Marano, F., & Monti, J. (2013). Knowledge management and Cultural Heritage repositories: Cross-Lingual Information Retrieval strategies. In *Digital Heritage International Congress (DigitalHeritage), 2013* (Vol. 2, pp. 295-302). IEEE.

di Buono, M. P., Monti, J., Monteleone, M., & Marano, F. (2013b). Multi-word processing in an ontology-based Cross-Language Information Retrieval model for specific domain collections. In *Workshop Proceedings for: Multi-word Units in Machine Translation and Translation Technologies*. The European Association for Machine Translation.

di Buono, M.P. & Monteleone, M. (2014). From Natural Language to Ontology Population in the Cultural Heritage Domain. A Computational Linguistics-based approach. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (LREC'14).

di Buono, M. P., Monteleone, M., & Elia, A. (2014a). Terminology and Knowledge Representation. Italian Linguistic Resources for the Archaeological Domain. In *Workshop on Lexical and Grammatical Resources for Language Processing* (p. 24-29) Coling 2014. ACL Web Anthology.

di Buono, M. P., Monteleone, M., & Elia, A. (2014b). How to Populate Ontologies. In *Natural Language Processing and Information Systems* (pp. 55-58). Springer International Publishing.

Di Sciullo, A. M., & Williams, E. (1987). *On the definition of word* (Vol. 14). Cambridge, MA: MIT press.

Doerr, M. (2003). The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. *AI magazine*, *24*(3), 75.

Dowty, D. (1991). Thematic proto-roles and argument selection. *language*, 547-619.

Dras, M. (1995). Automatic identification of support verbs: A step towards a definition of semantic weight. *arXiv preprint cmp-lg/9510007*.

Duchier, D., & Debusmann, R. (2001, July). Topological dependency trees: A constraint-based account of linear precedence. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (pp. 180-187). Association for Computational Linguistics.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis* (Vol. 3). New York: Wiley.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters.

Ekbal, A., Bonin, F., Saha, S., Stemle, E., Barbu, E., Cavulli, F., ... & Poesio, M. (2011). Rapid Adaptation of NE Resolvers for Humanities Domains using Active Annotation. *JLCL*, *26*(2), 39-51.

Elia, A. (1984). *Le verbe italien: les complétives dans les phrases à un complément*. Schena; Nizet.

Elia, A., Martinelli M., & D'Agostino E. (1981). *Lessico e strutture sintattiche.* Liguori*.*

Elia, A., Postiglione, A., & Monteleone, M. (2010). Cataloga. Sistema informatico per la catalogazione automatica di testi. Release 4.8. Software

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179-211.

Elsayed, A., El-Beltagy, S. R., Rafea, M., & Hegazy, O. (2007). Applying data mining for ontology building. *Proc. of ISSR*.

El-Sonbaty, Y., & Ismail, M. A. (1998). Fuzzy clustering for symbolic data. *Fuzzy Systems, IEEE Transactions on*, *6*(2), 195-204.

Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2004). Open information extraction from the web. *Communications of the ACM*, *51*(12), 68-74.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A. M., Shaked, T., Soderland, S., ... & Yates, A. (2004). Methods for domain-independent information extraction from the web: an experimental comparison. In *Proceedings of the 19th national conference on Artifical intelligence* (pp. 391-398). AAAI Press.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., & Mausam, M. (2011). Open Information Extraction: The Second Generation. In *IJCAI* (Vol. 11, pp. 3-10).

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1535-1545). Association for Computational Linguistics.

Falk, Y. (2001). *Lexical-functional grammar*. CSLI.

Fang, J., Guo, L., Wang, X., & Yang, N. (2007). Ontology-based automatic classification and ranking for web documents. In *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on* (Vol. 3, pp. 627-631). IEEE.

Fanizzi, N., d'Amato, C., & Esposito, F. (2008). DL-FOIL concept learning in description logics. In *Inductive Logic Programming* (pp. 107-121). Springer Berlin Heidelberg.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Fernie, K., Gavrilis, D., & Angelis, S. (2013). *The CARARE metadata schema, v. 2.0*. CARARE Tech. Rep.

Firth J.P. 1957. Papers in linguistics, 1934-1951. Oxford University Press, 1957

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine learning*, *2*(2), 139-172.

Flickinger, D., Pollard, C., & Wasow, T. (1985). Structure-sharing in lexical representation. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics* (pp. 262-267). Association for Computational Linguistics.

Florian, R., Ittycheriah, A., Jing, H., & Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (pp. 168-171). Association for Computational Linguistics.

Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The computer journal*, *41*(8), 578-588.

Freckleton, Peter (1985). Sentence idioms in English. Working Papers in Linguistics 11, University of Melbourne, pp. 153-168+ appendix (196 p.), ISSN 1443-6914.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *ICML* (Vol. 96, pp. 148-156).

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119-139.

Frické, M. (2009). The knowledge pyramid: a critique of the DIKW hierarchy. *Journal of information science*, *35*(2), 131-142.

Gangemi, A. (2013). A comparison of knowledge extraction tools for the semantic web. In *The semantic web: Semantics and big data* (pp. 351-366). Springer Berlin Heidelberg.

Gardent, C. (2006).Tree Adjoining Grammars. Theory and Practice. Bangkok. http://www.loria.fr/~gardent/teaching/tag-bkk06.pdf.

Garside, R. (1988). *The computational analysis of English: A corpus-based approach* (Vol. 57). G. Sampson, & G. Leech (Eds.). Longman.

Gazdar, G. (1982). Phrase structure grammar. In *The nature of syntactic representation* (pp. 131-186). Springer Netherlands.

Gazdar, G. (1985). *Generalized phrase structure grammar*. Harvard University Press.

Giuglea, A. M., & Moschitti, A. (2006). Semantic role labeling via framenet, verbnet and propbank. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 929-936). Association for Computational Linguistics.

Glass, James R., Timothy J. Hazen (1998). Telephone-based conversational speech recognition in the Jupiter domain, *Proceedings of the International Conference on Spoken Language Processing,* Sydney, Australia.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 513-520).

Goddard, C. (2011). *Semantic analysis: A practical introduction*. Oxford University Press.

Gómez-Pérez, A., & Manzano-Macho, D. (2003). A survey of ontology learning methods and techniques. *OntoWeb Deliverable D*, *1*(5).

Gorea, D., & Buraga, S. (2006). Towards integrating decision tree with xml technologies. In *Proceedings of the 8th International Conference on Development and Application Systems–DAS*.

Grąbczewski, K. (2014). *Meta-Learning in Decision Tree Induction*. Springer.

Gracia, J., Montiel-Ponsoda, E., & Gómez-Pérez, A. (2012). Cross-lingual Linking on the Multilingual Web of Data (position statement).

Gracia, J., Montiel-Ponsoda, E., Cimiano, P., Gómez-Pérez, A., Buitelaar, P., & McCrae, J. (2012b). Challenges for the multilingual web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, *11*, 63-71.

Gradmann, S. (2010). Knowledge= Information in context: on the importance of semantic contextualisation in Europeana. White Paper.

Greenberg, J. H. (1963). Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, *2*, 73-113.

Grefenstette, G., & Teufel, S. (1995). Corpus-based method for automatic identification of support verbs for nominalizations. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics* (pp. 98-103). Morgan Kaufmann Publishers Inc..

Grishman, R., & Sundheim, B. (1996). Message Understanding Conference-6: A Brief History. In *COLING* (Vol. 96, pp. 466-471).

Gross, M. (1975). *Méthodes en syntaxe*. Hermann.

Gross, M. (1979). On the failure of generative grammar. *Language*, 859-885.

Gross, M. (1981). Les bases empiriques de la notion de prédicat sémantique. *Langages*, 7-52.

Gross, M. (1982). Une classification des phrases «figées» du français. *Revue québécoise de linguistique*, *11*(2), 151-185.

Gross, M. (1984). Lexicon-grammar and the syntactic analysis of French. In *Proceedings of the 10th International Conference on Computational linguistics* (pp. 275-282). Association for Computational Linguistics.

Gross, M. (1986a). Lexicon-grammar: the representation of compound words. In *Proceedings of the 11th coference on Computational linguistics* (pp. 1-6). Association for Computational Linguistics.

Gross, M. (1986b). *Grammaire transformationelle du français*. Vol. 1, Syntaxe du verbe. Paris: Larousse.

Gross, M. (1989). La construction de dictionnaires électroniques. In *Annales des télécommunications* (Vol. 44, No. 1-2, pp. 4-19). Springer-Verlag.

Gross, M., Halle M. & Schützenberger M.P. (1973). Formal analysis of natural languages. *Proceedings of the first international conference* (Paris 1970). Paris: The Hague.

Gross, M., Halle, M., & Schützenberger, M. P. (1973). The formal analysis of natural languages. *Mouton, La Haye et Paris*.

Gross, M. (1997). 11 The Construction of Local Grammars. *Finite-state language processing*, 329.

Grosso, W. E., Eriksson, H., Fergerson, R. W., Gennari, J. H., Tu, S. W., & Musen, M. A. (1999). Knowledge modeling at the millennium. *Proc. KAW'99*.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, *5*(2), 199-220.

Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy* (Vol. 46). IOS press.

Halverson, A., Burger, J., Galanis, L., Kini, A., Krishnamurthy, R., Rao, A. N., ... & DeWitt, D. J. (2003). Mixed mode XML query processing. In *Proceedings of the 29th international conference on Very large data bases-Volume 29* (pp. 225-236). VLDB Endowment.

Han, J., & Kamber, M. (2001). Data mining: concept and technology. *Publishing House of Mechanism Industry*, 70-72.

Handschuh, S. (2005). *Creating ontology-based metadata by annotation for the semantic web* (Doctoral dissertation, Karlsruhe, Univ., Diss., 2005).

Hanks, P. (2013). *Lexical analysis: Norms and exploitations*. Mit Press.

Harman, G. H. (1963). Generative Grammars without Transformation Rules: A Defense of Phrase Structure. *Language*, 597-616.

Harper, M. P. & Helzerman, R. A. (1995). Extensions to constraint dependency parsing for spoken language processing. *Computer Speech & Language*, *9*(3), 187-234.

Harris, Z. S. (1951). *Methods in Structural Linguistics*. Chicago: University of Chicago Press.

Harris, Z. S. (1952). Discourse analysis: a sample text. *Language*, *28*(4), 1.

Harris, Z. S. (1954). Distributional structure. *Word*., X/2-3, 1954. 146-62 [reprinted in Harris Zellig S. 1970. Papers in Structural and Transformational Linguistics. Dordrecht: Reidel. 775-794].

Harris, Z. S. (1964). Transformations in linguistic structure. *Proceedings of the American Philosophical Society*, 418-422.

Harris, Z. S. (1968). Mathematical structures of language. Wiley, New York.

Harris, Z. S. (1970). Papers in Structural and Transformational Linguistics. Formal Linguistics Series, Volume 1.

Harris, Z. S. (1976). *Notes de Cours de Syntaxe: Traduit de l'anglais par Maurice Gross*. Seuil.

Harris, Z. S. (1982). *A grammar of English on mathematical principles*. John Wiley & Sons Inc.

Harris, Z. S. (1988). *Language and information*. Columbia University Press.

Harris, Z.S. (1946). From Morpheme to Utterance. *Language* 22:3.161–183.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83-85.

Hays, D. G. (1964). Dependency theory: A formalism and some observations. *Language*, 511-525.

Hazman, M., El-Beltagy, S. R., & Rafea, A. (2011). A survey of ontology learning approaches. *database*, *7*, 6.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2* (pp. 539-545). Association for Computational Linguistics.

Hecht-Nielsen, R. (1989). Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on* (pp. 593-605). IEEE.

Hellmann, S., Lehmann, J., Auer, S., & Brümmer, M. (2013). Integrating nlp using linked data. In *The Semantic Web–ISWC 2013* (pp. 98-113). Springer Berlin Heidelberg.

Hellwig, P. (1986). Dependency unification grammar. In *Proceedings of the 11th coference on Computational linguistics* (pp. 195-198). Association for Computational Linguistics.

Hjørland, B. (2007). Semantics and knowledge organization. *Annual review of information science and technology*, *41*(1), 367-405.

Hobbs, J. R. (1993). The generic information extraction system. In *MUC* (pp. 87-91).

Horácek, P., Zámecnıková, E., & Burgetová, I. (2011). Generalized Phrase Structure Grammar. http://www.fit.vutbr.cz/~rudolfa/grants.php?file=%2Fproj%2F533%2Ffmnl04-gpsg.pdf&id=533

Hourali, M., & Montazer, G. A. (2012). Using ART2 Neural Network and Bayesian Network for Automating the Ontology Constructing Process. *Procedia Engineering*, *29*, 3914-3923.

Hudson, R. A. (1984). *Word grammar*. Oxford: Blackwell.

Hudson, R. A. (1990). *English word grammar* (Vol. 108). Oxford: Basil Blackwell.

Humbley, J. (1997). Is terminology specialized lexicography? The experience of Frenchspeaking countries. *Hermès*, 13-31.

Hurford, M., Harabagiu, S., Williams, J., & Aarseth, P. (2003). Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 8-15). Association for Computational Linguistics.

Ide, N., & Pustejovsky, J. (2010). What does interoperability mean, anyway. *Toward an operational definition of interoperability*.

Jackendoff, R. (1977). X syntax: A study of phrase structure. *Linguistic Inquiry Monographs Cambridge, Mass.*, (2), 1-249.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc..

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.

Jarvinen, T., & Tapanainen, P. (1998). Towards an implementable dependency grammar. In *Proceedings of the Workshop on Processing of Dependency-Based Grammars* (Vol. 10).

Jones, M. N., & Mewhort, D. J. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, *114*(1), 1.

Joshi, A. K., Levy, L. S., & Takahashi, M. (1975). Tree adjunct grammars. *Journal of computer and system sciences*, *10*(1), 136-163.

Jurafsky, D., & Martin, J. H. (2009). *Speech & language processing*. 2nd edition. Prentice-Hall..

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 237-285.

Kahane, S. (2003). The meaning-text theory. *Dependency and Valency. An International Handbook of Contemporary Research*, *1*, 546-570.

Karampinas, D., & Triantafillou, P. (2012). Crowdsourcing taxonomies. In *The semantic web: Research and applications* (pp. 545-559). Springer Berlin Heidelberg.

Karlsson, F. (1990). Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th conference on Computational linguistics-Volume 3* (pp. 168-173). Association for Computational Linguistics.

Karlsson, F., Voutilainen, A., Heikkilae, J., & Anttila, A. (Eds.). (1995). *Constraint Grammar: a language-independent system for parsing unrestricted text* (Vol. 4). Walter de Gruyter.

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, 119-127.

Kipper, K., Dang, H. T., & Palmer, M. (2000). Class-based construction of a verb lexicon. In *AAAI/IAAI* (pp. 691-696).

Kipper, K., Korhonen, A., Ryant, N., & Palmer, M. (2006). Extending VerbNet with novel verb classes. In *Proceedings of LREC* (Vol. 2006, No. 2.2, p. 1).

Kipper, K., Palmer, M., & Rambow, O. (2002). Extending propbank with verbnet semantic predicates. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas (AMTA-2002), Tiburon, CA, USA, October* (pp. 6-12).

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 423-430). Association for Computational Linguistics.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, *51*(3), 455-500.

Konstantinova, N. (2014). Review of Relation Extraction Methods: What Is New Out There?. In *Analysis of Images, Social Networks and Texts* (pp. 15-28). Springer International Publishing.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Kruijff, G.-J. M. (2001). *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic and Information Structure.* PhD thesis, Charles University.

Kübler, S., McDonald, R., & Nivre, J. (2009). Dependency parsing. *Synthesis Lectures on Human Language Technologies*, *1*(1), 1-127.

Lakatos, I. (1978). The methodology of scientific research programmes, vol. 1. *Philosophical Papers. Cambridge: Cambridge University Press*.

Lakshmi, T. M., Martin, A., Begum, R. M., & Venkatesan, V. P. (2013). An analysis on performance of decision tree algorithms using student's qualitative data. *International Journal of Modern Education and Computer Science (IJMECS)*, *5*(5), 18.

Laporte, E. (2005). In memoriam Maurice Gross. In *Archives of Control Sciences* (Vol. 15, No. 3, pp. 257-278).

Lehmann, J., & Hitzler, P. (2010). Concept learning in description logics using refinement operators. *Machine Learning*, *78*(1-2), 203-250.

Lehmann, J. & Völker, J. (2014), An Introduction to Ontology Learning, in Jens Lehmann & Johanna Völker, ed., 'Perspectives on Ontology Learning' , AKA / IOS Press, , pp. ix-xvi .

Lempel R. (2010) Introduction to Search Engine Technology. Term-at-a-Time and Document-at-a-Time Evaluation. http://webcourse.cs.technion.ac.il/236621/Winter2010-2011/ho/WCFiles/lec4-evaluation.pdf.

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33-38.

Lenat, D. B., Prakash, M., & Shepherd, M. (1985). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI magazine*, *6*(4), 65.

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *From context to meaning: Distributional models of the lexicon in linguistics and cognitive science, special issue of the Italian Journal of Linguistics*, *20*(1), 1-31.

Bloomfield L. (1993). Language. University of Chicago Press, 1933

Levin, B. (1993). *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Levine, R. D., & Meurers, W. D. (2006). Head-driven phrase structure grammar. *Encyclopedia of Language and Linguistics,*, 237-252.

Li F. (2015). Event extraction in MUC. *Lecture of Internet-based IE technologies.* http://www.cs.sjtu.edu.cn/~li-fang/Lecture%206-7%20Event%20IE.pdf

Lin, D., & Pantel, P. (2001). DIRT@ SBT@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 323-328). ACM.

Loh, W. Y., & Shih, Y. S. (1997). Split selection methods for classification trees. *Statistica sinica*, *7*(4), 815-840.

Luuk, E. (2009). The noun/verb and predicate/argument structures. *Lingua*, *119*(11), 1707-1727.

Machonis, Peter A. (1985). Transformations of verb phrase idioms: passivization, particle movement, dative shift. *American Speech* 60:4, pp. 291-308.

Mädche, A., & Volz, R. (2001). The ontology extraction & maintenance framework Text-To-Onto. In *Proc. Workshop on Integrating Data Mining and Knowledge Management, USA* (pp. 1-12).

Magerman, D. M. (1995). Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* (pp. 276-283). Association for Computational Linguistics.

Maimon, O., & Rokach, L. (Eds.). (2005). *Data mining and knowledge discovery handbook* (Vol. 2). New York: Springer.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1, p. 496). Cambridge: Cambridge university press.

Marano, F. (2012) *Exploring Formal Models of Linguistic Data Structuring. Enhanced Solutions for Knowledge Management Systems Based on NLP Applications*. PhD Dissertation, University of Salerno, Italy.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, *19*(2), 313-330.

Maruyama, H. (1990). *Constraint dependency grammar*. Technical Report# RT0044, IBM, Tokyo, Japan.

Mascardi, V., Cordì, V., & Rosso, P. (2007, September). A Comparison of Upper Ontologies. In *WOA* (pp. 55-64).

Matwin, S., & Szpakowicz, S. (1993). Text analysis: how can machine learning help?. In *Proceedings of the 1st Conference of the Pacific Association for Computational Linguistics (PACLING)* (pp. 33-42).

McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (pp. 188-191). Association for Computational Linguistics.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, pp. 41-48).

McCarthy, J. (1980). Circumscription| a form of non-monotonic reasoning. *Artificial Intelligence*, 13, 27-39. Reprinted in Readings in Artificial Intelligence (BL Webber and NJ Nilsson, eds.).

McDermott, D., & Doyle, J. (1980). Non-monotonic logic I. *Artificial intelligence*, *13*(1), 41-72.

McDowell, L. K., & Cafarella, M. (2008). Ontology-driven, unsupervised instance population. *Web Semantics: Science, Services and Agents on the World Wide Web*, *6*(3), 218-236.

Mel'čuk, I. A., & Polguère, A. (1987). A formal lexicon in the meaning-text theory:(Or how to do lexica with words). *Computational linguistics*, *13*(3-4), 261-275.

Mendel, T. (2013). Business Intelligence and Big Data Trends. http://www.hfsresearch.com/report/business-intelligence-and-big-data-trends-2013.

Menzel, W., & Schröder, I. (1998). Decision procedures for dependency parsing using graded constraints. In *in proceedings of ACL'90*.

Mikolov, T. (2012). Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010* (pp. 1045-1048).

Mikolov, T., Yih, W. T., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL* (pp. 746-751).

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39-41.

Mirambicka, R., Sulthana, A. R., & Vadivu, G. (2013). Decision Tree Applied to Learning Relations between Ontologies. *Lecture Notes on Software Engineering*, *1*(2), 164.

Mirambicka, R., Sulthana, A. R., & Vadivu, G. (2013). Decision Tree Applied to Learning Relations between Ontologies. *Lecture Notes on Software Engineering*, *1*(2), 164.

Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, *45*.

Mitkov, R. (1999). *Anaphora resolution: the state of the art*. School of Languages and European Studies, University of Wolverhampton.

Montague, R. (1970). English as a formal language.

Monti, J., Monteleone, M., di Buono, M. P., & Marano, F. (2013). Natural Language Processing and Big Data-An Ontology-Based Approach for Cross-Lingual Information Retrieval. In *Social Computing (SocialCom), 2013 International Conference on* (pp. 725-731). IEEE.

Moore, R. (1987). Possible-world semantics for autoepistemic logic. In *Readings in nonmonotonic reasoning* (pp. 137-142). Morgan Kaufmann Publishers Inc..

Moro, A., & Navigli, R. (2013). Integrating syntactic and semantic analysis into the open information extraction paradigm. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence* (pp. 2148-2154). AAAI Press.

Morris, J., & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*, *17*(1), 21-48.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Muslea, I. (1999). Extraction patterns for information extraction tasks: A survey. In *The AAAI-99 Workshop on Machine Learning for Information Extraction* (Vol. 2, No. 2).

Nevin, B. E., & Johnson, S. M. (2002). The Legacy of Zellig Harris. *Language and information into the 21st century (Amsterdam studies in the theory and history of linguistic science). Amsterdam: J. Benjamins Pub*, *228*, 229.

Nezhadi, A. H., Shadgar, B., & Osareh, A. (2011). Ontology alignment using machine learning techniques. *International Journal of Computer Science & Information Technology*, *3*(2), 139.

Nguyen, A., Yosinski, J., & Clune, J. (2014). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv preprint arXiv:1412.1897*.

Nivre, J. (2005). Dependency grammar and dependency parsing. *MSI report*, *5133*(1959), 1-32. http://stp.lingfil.uu.se/~nivre/docs/05133.pdf

Noy, N. F., Fergerson, R. W., & Musen, M. A. (2000). The knowledge model of Protege-2000: Combining interoperability and flexibility. In *Knowledge Engineering and Knowledge Management Methods, Models, and Tools* (pp. 17-32). Springer Berlin Heidelberg.

Obaseki, T. I. (2010). Automated Indexing: The Key to Information Retrieval in the 21st Century. *Library Philosophy and Practice*.

Ogden, C. K., & Richards, I. A. (1923). The Meaning of Meaning. Reprinted in 1989. Harvest/HBJ.

Omelayenko, B. (2001). Learning of ontologies for the web: the analysis of existent approaches. In *First International Workshop on Web Dynamics in Conjunction with the Eighth International Conference on Database Theory London, UK* (p. 16).

Ong, E., Hong, B. A., & Nuñez, V. A. (2008). Automatically Extracting Templates from Examples for NLP Tasks. In *PACLIC* (pp. 452-459).

Oren, E., Delbru, R., Möller, K., Völkel, M., & Handschuh, S. (2006). Annotation and Navigation in Semantic Wikis. In *Proceedings of the Workshop on Semantic Wikis (SemWiki), in conjunction with 3rd European Semantic Web Conference*. NUI Galway.

Orphanos, G., Kalles, D., Papagelis, A., & Christodoulakis, D. (1999). Decision trees and NLP: A case study in POS tagging. In *Proceedings of annual conference on artificial intelligence (ACAI)*.

Osborne, T. (2013). A Look at Tesnière's Éléments through the Lens of Modern Syntactic Theory. *DepLing 2013*, 262.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, *31*(1), 71-106.

Pasca, M. (2004). Acquisition of categorized named entities for web search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management* (pp. 137-145). ACM.

Pease, A., Niles, I., & Li, J. (2002). The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *Working notes of the AAAI-2002 workshop on ontologies and the semantic web* (Vol. 28).

Peng, Y. (2010). *Ontology mapping neural network: An approach to learning and inferring correspondences among ontologies* (Doctoral dissertation, University of Pittsburgh).

Pollard, C., & Sag, I. (1987). Information-based Syntax and Semantics, vol. 1.

Pollard, C., & Sag, I. A. (1994). *Head-driven phrase structure grammar*. University of Chicago Press.

Proudian, D., & Pollard, C. (1985). Parsing head-driven phrase structure grammar. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics* (pp. 167-171). Association for Computational Linguistics.

Putthividhya, D. P., & Hu, J. (2011). Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1557-1567). Association for Computational Linguistics.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81-106.

Quinlan, J. R. (1996). Improved use of continuous attributes in C4. 5. *Journal of artificial intelligence research*, 77-90.

Raikwal, J. S., & Saxena, K. (2012). Performance evaluation of SVM and k-nearest neighbor algorithm over medical data set. *International Journal of Computer Applications*, *50*(14).

Raschka Sebastian (2014) Naive Bayes and Text Classication. Introduction and Theory. Available at http://arxiv.org/pdf/1410.5329.pdf.

Ravuri, S., & Stolcke, A. (2014). Neural Network Models for Lexical Addressee Detection. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Reiter, R. (1980). A logic for default reasoning. *Artificial intelligence*, *13*(1), 81-132.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence* (pp. 1044-1049).

Riloff, E., & Jones, R. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI* (pp. 474-479).

Ritter, A., Clark, S., & Etzioni, O. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1524-1534). Association for Computational Linguistics.

Rojas, R. (2009). AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting. *Freie University, Berlin, Tech. Rep*.

Rokach, L., & Maimon, O. (2005). Decision trees. In Data Mining and Knowledge Discovery Handbook (pp. 165-192). Springer US.

Rokach, L., & Maimon, O. (2005b). Clustering Methods. In Data Mining and Knowledge Discovery Handbook. Springer US.

Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Rowley, J. (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science*, *33*(2), 163-180.

Ruppenhofer, J., Ellsworth, M., Petruck, M. R., Johnson, C. R., & Scheffczyk, J. (2006). FrameNet II: Extended theory and practice.

Sabou, M., Wroe, C., Goble, C., & Mishne, G. (2005, May). Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In *Proceedings of the 14th international conference on World Wide Web* (pp. 190-198). ACM.

Sag, I. A., Kaplan, R., Karttunen, L., Kay, M., Pollard, C., Shieber, S. M., & Zaenen, A. (1986). Unification and grammatical theory. http://lingo.stanford.edu/sag/papers/sagetal-1986.pdf

Salton, G. (1971). The SMART retrieval system—experiments in automatic document processing.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513-523.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.

Sánchez Cisneros, D., & Aparicio Gali, F. (2015). Uem-uc3m: An ontology-based named entity recognition system for biomedical texts. Association for Computational Linguistics.

Sánchez Cisneros, D., & Moreno, A. (2004). Creating ontologies from Web documents. *Recent Advances in Artificial Intelligence Research and Development. IOS Press*, *113*, 11-18.

Saussure, de, F. (1995). Cours de linguistique générale (1922) Recueil des publications scientifiques. Bally, C., and Gautier, L. (eds.) Lausanne and Geneva: Payot.

Schabes, Y. (1990). Mathematical and computational aspects of lexicalized grammars. PhD thesis, University of Pennsylvania,

Schapire, R. E. (2013). Explaining adaboost. In *Empirical inference* (pp. 37-52). Springer Berlin Heidelberg.

Schmitz, M., Bart, R., Soderland, S., & Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 523-534). Association for Computational Linguistics.

Schröder, I. (2002). Natural Language Parsing with Graded Constraints. PhD thesis, Hamburg University.

Schuler, K. K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon.

Schwenk, H. (2007). Continuous space language models. *Computer Speech & Language*, *21*(3), 492-518.

Sgall, P., Hajicová, E., & Panevová, J. (1986). *The meaning of the sentence in its semantic and pragmatic aspects*. Springer Science & Business Media.

Shamsfard, M., & Abdollahzadeh Barforoush, A. (2003). The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, *18*(04), 293-316.

Shamsfard, M., & Barforoush, A. A. (2004). Learning ontologies from natural language texts. *International journal of human-computer studies*, *60*(1), 17-63.

Shang, W., Zhu, H., Huang, H., Qu, Y., & Lin, Y. (2006). The improved ontology KNN algorithm and its application. In *Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on* (pp. 198-203). IEEE.

Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell system technical journal*, *30*(1), 50-64.

Shieber, S. M. (1986). *An introduction to unification-based approaches to grammar*. CSLI Publications.

Shiffman, D. (2012) The Nature of Code: Simulating Natural Systems with Processing. Paperback.

Shriberg, L. D., & Kwiatkowski, J. (1994). Developmental Phonological Disorders IA Clinical Profile. *Journal of Speech, Language, and Hearing Research*, *37*(5), 1100-1126.

Silberztein, M. (1993). *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*. Masson.

Silberztein, M. (2003). NooJ manual. *Available for download at: www. nooj-association.org*.

Silberztein, M. (2015). La formalisation des langues : l'approche de NooJ. ISTE: London.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 21-29). ACM.

Slimani, T. (2013). Description and evaluation of semantic similarity measures approaches. *arXiv preprint arXiv:1310.8059*.

Sneath, P. H., & Sokal, R. R. (1973). Numerical taxonomy. The principles and practice of numerical classification.

Socher, R., Lin, C. C., Manning, C., & Ng, A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 129-136).

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (Vol. 1631, p. 1642).

Šochman J., Jiří Matas, Jana Kostlivá, Ondřej Drbohlav. (2014). AdaBoost. Center for Machine Perception, Czech Technical University, Prague. https://cw.fel.cvut.cz/wiki/_media/courses/a4b33rpz/adaboost_talk_lecture2014_11_14.pdf.

Sohn, J. S., Wang, Q., & Chung, I. J. (2013). Generation of User Interest Ontology Using ID3 Algorithm in the Social Web. In *IT Convergence and Security 2012* (pp. 1067-1074). Springer Netherlands.

Sowa, J. F. (1999). Knowledge representation: logical, philosophical, and computational foundations. Pacific Grove, CA: Brooks Cole Publishing Co.

Sowa, J. F. (2001). Building, sharing, and merging ontologies. *web site: http://www. jfsowa. com/ontology/ontoshar. htm*.

Staab, S., & Mädche, A. (2001). Ontology learning for the semantic web. *IEEE Intelligent Systems: Special Issue on the Semantic Web*.

Starosta, S. (1988). *The case for lexicase: an outline of lexicase grammatical theory*. Pinter Pub Limited.

Steedman, M. (2000). *The syntactic process* (Vol. 24). Cambridge: MIT press.

Steedman, M., & Baldridge, J. (2011). Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell*.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., & Wenke, D. (2002). *OntoEdit: Collaborative ontology development for the semantic web* (pp. 221-235). Springer Berlin Heidelberg.

Sureshkumar, G., & Zayaraz, G. (2015). Automatic relation extraction using naïve Bayes classifier for concept relational ontology development. *International Journal of Computer Aided Engineering and Technology*, *7*(4), 421-435.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management* (pp. 67-74). ACM.

Tapanainen, P., & Järvinen, T. (1997). A non-projective dependency parser. In *Proceedings of the fifth conference on Applied natural language processing* (pp. 64-71). Association for Computational Linguistics.

Taylor, A., Marcus, M., & Santorini, B. (2003). The Penn treebank: an overview. In *Treebanks* (pp. 5-22). Springer Netherlands.

Tesnière, L. (1959). *Eléments de syntaxe structurale*. Librairie C. Klincksieck.

Teufel S. (2014) Lecture 1: Introduction and Overview. Information Retrieval Computer Science Tripos Part II, http://www.cl.cam.ac.uk/teaching/1314/InfoRtrv/lecture1.pdf.

Todorov K. (2006). Aspects of Learning Ontologies With Support Vector Machines. *Presentation at 27th Annual Meeting of the DGfS 2006. http://tcl.sfs.unituebingen.de/tt/Workshop_Materialien/Todorov_SVMs.pdf*

Tur, G., Deng, L., Hakkani-Tür, D., & He, X. (2012). Towards deeper understanding: deep convex networks for semantic utterance classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on* (pp. 5045-5048). IEEE.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384-394). Association for Computational Linguistics.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *J. of Math*, *58*(345-363), 5.

Turing, A.M. (1939). Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, *2*(1), 161-228.

Turney, P. D. (2008). The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 615-655.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*(1), 141-188.

Van Damme, C., Hepp, M., & Siorpaes, K. (2007). Folksontology: An integrated approach for turning folksonomies into ontologies. *Bridging the Gap between Semantic Web and Web*, *2*(2), 57-70.

Vander Wal, T. (2005). Off the Top: Folksonomy Entries. Visited November 5, 2005. http://vanderwal.net/folksonomy.html

Vapnik, V. N., & Chervonenkis, A. J. (1974). Theory of pattern recognition.

Vargas-Vera, M., Motta, E., Domingue, J., Shum, S. B., & Lanzoni, M. (2001). Knowledge Extraction by Using an Ontology Based Annotation Tool. In *Semannot@ K-CAP 2001*.

Vassev, E., & Hinchey, M. (2011). Knowledge representation and reasoning for intelligent software systems. *Computer*, (8), 96-99.

Velardi, P., Navigli, R., Cuchiarelli, A., & Neri, R. (2005). Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies.

Ontology learning from text: Methods, applications and evaluation, *123*, 92-106.

Vietri, S. (2014) The italian module for NooJ. *Proceedings of the First Italian Conference on Computational Linguistics*, CLiC-it 2014. Pisa University Press.

Vietri, S., & Monteleone, M. (2014) The English NooJ dictionary. In *Proceedings of NooJ 2013 International Conference, June* (pp. 3-5).

Vijay-Shanker, K. (1992). Using descriptions of trees in a tree adjoining grammar. *Computational Linguistics*, *18*(4), 481-517.

Vijay-Shanker, K., & Joshi, A. K. (1988). Feature structures based tree adjoining grammars. In *Proceedings of the 12th conference on Computational linguistics-Volume 2* (pp. 714-719). Association for Computational Linguistics.

Völker, J. (2009). *Learning expressive ontologies* (Vol. 2). IOS Press.

Völker, J., & Niepert, M. (2011). Statistical schema induction. In *The Semantic Web: Research and Applications* (pp. 124-138). Springer Berlin Heidelberg.

Weibel, S., Kunze, J., Lagoze, C., & Wolf, M. (1998). *Dublin core metadata for resource discovery* (No. RFC 2413).

Wittgenstein, L. (1953). Philosophical investigations. (Anscombe, G.E.M., trans.). Oxford: Basil Blackwell.

Wu, F., & Weld, D. S. (2010). Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 118-127). Association for Computational Linguistics.

Wuermli, O., Wrobel, A., Hui, S. C., & Joller, J. M. (2003). *Data Mining For Ontology Building: Semantic Web Overview* (Doctoral dissertation, Diploma Thesis–Dep. of Computer Science WS2002/2003, Nanyang Technological University).

XTAG Research Group. (2001). XTAG Technical Report. *University of Pennsylvania, Uppen*, *29*.

Yao, L., Haghighi, A., Riedel, S., & McCallum, A. (2011). Structured relation discovery using generative models. In *Proceedings of the Conference*

on Empirical Methods in Natural Language Processing (pp. 1456-1466). Association for Computational Linguistics.

Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., & Soderland, S. (2007). Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations* (pp. 25-26). Association for Computational Linguistics.

Yuen, D., & Koehler-Kruener, H. (2012). Who's Who in Text Analytics. *Stamford, CT: Gartner, Inc*.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *Computers, IEEE Transactions on*, *100*(1), 68-86.

Zavrel, J., Daelemans, W., & Veenstra, J. (1997). Resolving PP attachment ambiguities with memory-based learning. In *Proc. of the Workshop on Computational Language Learning (CoNLL'97), ACL, Madrid*.

Zayaraz, G. (2015). Concept relation extraction using Naïve Bayes classifier for ontology-based question answering systems. *Journal of King Saud University-Computer and Information Sciences*, *27*(1), 13-24.

Zhang, D., & Lee, W. S. (2004). Web taxonomy integration using support vector machines. In *Proceedings of the 13th international conference on World Wide Web* (pp. 472-481). ACM.

Zhou, Z. H. (2012). *Ensemble methods: foundations and algorithms*. CRC Press.

Zhu, J., Nie, Z., Liu, X., Zhang, B., & Wen, J. R. (2009). StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World Wide Web* (pp. 101-110). ACM.

Zouaq, A. (2008). Une approche d`ingenierie ontologique pour l`acquisition et l'exploitation des connaissances a partir de documents textuels : Vers des objets de connaissances et d`apprentissage, PhD dissertation, Department of Computer Science and Operations Research, University of Montreal, 2008.

Zouaq, A. (2011). An overview of shallow and deep natural language processing for ontology learning. *Ontology learning and knowledge discovery using the web: Challenges and recent advances*, *2*, 16-37.