Università degli Studi di Salerno

Dipartimento di Informatica

Dottorato di Ricerca in Informatica

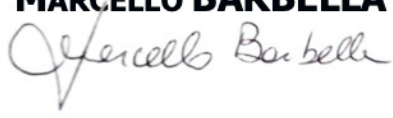
Curriculum Internet of Things and Smart Technologies

XXXV Ciclo

TESI DI DOTTORATO / PH.D. THESIS

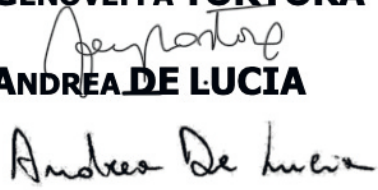
Data Integration and Automatic Text Summarization: A path to more informed Business Decisions

MARCELLO BARBELLA



SUPERVISOR: **PROF. GENOVEFFA TORTORA**

PHD PROGRAM DIRECTOR: **PROF. ANDREA DE LUCIA**



A.A 2021/2022

To my mother,
she that taught me to laugh, to hope and to pray.

They will tell you that you are not enough.
Don't be fooled.
You are much better than what they want you to believe.

(John Paul II)

Dedicated to the affectionate memory of Prof. Michele Risi.

You can not connect the dots by looking to the future;
you can only connect them by looking at the past.
You have to trust that the dots will connect,
somehow, in the future.
You have to have faith in something:
your intuition, destiny, life, karma, whatever...
This approach has never beaten me down,
and it has made all the difference in my life.

Steve Jobs

ACKNOWLEDGMENTS

Like any train that leaves from a station and arrives at its destination, my three-year PhD path has reached its conclusion. I can state, with certainty, that this road served as a very significant parenthesis in my life, both for the numerous challenges I had to face and overcome, as well as for my personal and cultural development. The first of these obstacles was something unreal, something that no one would ever have thought would come true, and that has radically changed the life of every human being in this world: the pandemic.

Only a few months had passed since I started my journey when, in March 2020, the students of the university, teachers, and staff received an email from the Rector, informing them that the university would be closing as a result of this unreal event. Reading the email felt like the end of the world to me personally, but I think it did to each of us as well. I had no idea that this day would mark the beginning of a seemingly endless period that would last until virtually the end of my adventure.

The pandemic has cost me a great deal. I missed out on the chance to regularly interact with my teachers, attend to my colleagues every day, and experience the university as I dreamed. But, despite this great difficulty, I am incredibly grateful to everyone I met and who, whether they were present in person or virtually, had a fundamental presence in my life. I would particularly like to thank Alessia Auriemma Citarella, who has been

a truly amazing find and a friend I will carry with me throughout life. Without her knowing me, I felt right at home from the moment I entered the lab. She proved to be a true friend, ready to help me, offer advice, and support me when I needed it most (and there were many moments), especially after the passing of my beloved tutor, my highly regarded prof. Michele Risi. His leaving has left me with an unfillable vacuum.

Thanks to him I have succeeded in accomplishing this path. I will not be able to forget his life lessons, discretion, special manner of being present without interfering, or his capacity to foresee objectives before they appeared on the horizon. *"Everyone must choose their path and put their all into whatever they meet"*. This is his core message, the turning point that captures the essence of his existence. A big hug for you.

There were other travellers with whom I shared my daily challenges and adventures. Each of them left me something. For those who find themselves spending days together, even a grin is vital, and they have given me so many.

I want to end by expressing my gratitude to Prof. Genoveffa Tortora. She was the first to have faith in me, to have confidence in my abilities, and to have made me feel deserving and capable of taking this route. She is now my current Tutor, and after Prof. Risi passed away, she took care of me by helping me evaluate and correct the many tasks I completed, giving me advice, and suggesting solutions and improvements. She even worked late into the night to support and encourage me. Not everyone is willing to give up their time, which is frequently taken away from their families, for others. But Prof. Tortora has made her work a mission, and she generously shares her enthusiasm with everyone privileged enough to come into contact with her, during their studies and life.

Last but not least, I want to thank my family for being there for me during every period of my life.

Marcello

If you can not be a pine on the top of the hill,
be a shrub in the valley, but be the best little shrub at the side of the hill...
Douglas Malloch

ABSTRACT

In recent years, there has been an explosion of data shared online. The majority of this internet information is in text format and can be used as a source to create new knowledge. These data are frequently unstructured and, in their raw state, it can be difficult to use them for the analyses, resulting in challenges to manage from an Information Technology (IT) perspective. But in addition to these types of data, most companies have a huge collection of structured data, acquired and built over time. The union of these two types of information represents therefore a gold mine to be able to draw as much knowledge as possible from them. Because of this, the so-called Data Pre-processing (DPP), an important stage in the Data Mining (DM) process, allows significant manipulations on them, in order to make them useable for any subsequent elaboration procedure. The general DPP steps are Data Cleansing, Data Integration, Data Reduction, and Data Transformation, while guaranteeing the protection of privacy. This research focused on two different applications related to structured and unstructured data, through respectively a focus on a Data Integration (DI) challenge, and one on the Automatic Text Summarization (ATS) task, whose algorithm evaluation metrics were explored.

One of the most challenging issues in DI, is the research for automatic or semi-automatic methodologies since these techniques often require the expertise of a domain specialist who can direct the process and improve the results. However, in the literature, there are not many fully or semi-automatic DI approaches unless they include experts with specific IT-skills. So, in this study, with the assistance of an intermediary figure (the *Company Manager*), who is not necessarily skilled in IT, using an Information Retrieval (IR) methodology, clustering methods, and a trained neural network, we have built a semi-automatic DI process. This process is capable of reducing persistent conflicts in data, and ensuring a unified view of them, respecting the original con-

straints of the datasets and guaranteeing a high-quality outcome for Business Intelligence evaluations.

At the same time, having the ability to reduce the amount of text from which to extract information is essential, when there are textual data sources involved. This is important to recover the key concepts, but also to speed up the analysis systems. In particular, ATS is an interesting challenge of Natural Language Processing (NLP). The primary issue is that there are currently several algorithms that attempt to reduce documents, using both mostly statistical techniques (Extractive algorithms) and Artificial Intelligence (AI) techniques (Abstractive algorithms). However, several metrics primarily based on the overlap analysis of *n-grams* such as ROUGE, which is the most used, are applied to assess the quality of the results. Therefore, determining if these metrics are efficient, and whether they enable us to compare the quality of the outcomes of the various ATS algorithms, is the focus of the second research topic.

ABSTRACT IN ITALIANO

Negli ultimi anni c'è stata un'esplosione dei dati condivisi online. La maggior parte di queste informazioni presenti su Internet è in formato testuale, e può essere utilizzata come fonte per produrre nuova conoscenza. Questi dati spesso non sono strutturati e, allo stato grezzo, non possono essere utilizzati per nessun tipo di analisi, risultando così difficili da gestire dal punto di vista dell'Information Technology (IT). Ma oltre a questi tipi di dati, la maggior parte delle aziende dispone di una vasta raccolta di dati strutturati, acquisiti e costruiti nel tempo. L'unione di queste due tipologie di informazioni rappresenta quindi una miniera d'oro per poterne trarre quanta più conoscenza possibile. Per tale motivo, il Data Pre-processing (DPP), una fase importante del processo di Data Mining (DM), consente importanti manipolazioni sugli stessi, al fine di renderli fruibili per eventuali elaborazioni successive. I passaggi generali del DPP sono la Pulizia, l'Integrazione, la Riduzione e la Trasformazione dei dati, garantendo nel contempo la protezione della privacy. Questo lavoro di ricerca si è concentrato su due diverse applicazioni

relative ai dati strutturati e non strutturati, attraverso rispettivamente un focus su Data Integration (DI) e uno sull'Automatic Text Summarization (ATS), di cui sono state esplorate le metriche di valutazione degli algoritmi.

Una delle sfide più impegnative per la DI, è la ricerca di metodologie completamente o parzialmente automatiche, poiché queste tecniche spesso richiedono l'esperienza di uno specialista del dominio, in grado di dirigere il processo e migliorarne i risultati. Tuttavia, in letteratura, non sono molti gli approcci di DI completamente o parzialmente automatici a meno che non includano esperti con specifiche competenze informatiche. Quindi, in questo studio, attraverso l'assistenza di una figura intermedia (il *Company Manager*), che non ha necessariamente competenze di Information Technology (IT), utilizzando una metodologia di Information Retrieval (IR), dei metodi di clustering e una rete neurale addestrata, abbiamo costruito un processo di DI semi-automatico. Esso è in grado sia di ridurre i conflitti persistenti nei dati, sia di garantire una visione unificata degli stessi, rispettando i vincoli originali dei dataset e fornendo un risultato di alta qualità per le valutazioni di Business Intelligence.

Allo stesso tempo, avere la capacità di ridurre la quantità di testo da cui estrarre informazioni è fondamentale, quando le fonti dati coinvolte sono testuali. Ciò è importante per recuperare i concetti chiave, ma anche per velocizzare i sistemi di analisi. In particolare, l'ATS è una sfida interessante del Natural Language Processing (NLP). Il problema principale è che attualmente esistono numerosi algoritmi che provano a riassumere i documenti, utilizzando sia tecniche per lo più statistiche (Algoritmi estrattivi) sia metodi di Artificial Intelligence (AI) (Algoritmi astrattivi). Tuttavia, per valutare la qualità dei risultati vengono utilizzate metriche basate principalmente sull'analisi della sovrapposizione di *n-grammi* come la ROUGE, che è quella più usata allo scopo. Pertanto, l'obiettivo del secondo argomento di ricerca è stato quello di determinare se tali metriche sono realmente efficienti e se consentono davvero di confrontare la qualità dei risultati dei vari algoritmi di ATS.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AATS Abstractive Automatic Text Summarization
AI Artificial Intelligence
ATS Automatic Text Summarization
BSS/TSS Between Sum of Squares/Total Sum of Squares
DI Data Integration
DL Deep Learning
DM Data Mining
DNN Deep Neural Network
DPP Data Pre-processing
DW Data Warehouse
EATS Extractive Automatic Text Summarization
ETL Extract Transform Load
FBE Frequency Based Embeddings
FNN Feed-forward Neural Network
GUI Graphical User Interface
GRU Gated Recurrent Unit
IoT Internet of Things
IR Information Retrieval
IT Information Technology
LD Levenshtein Distance
LSI Latent Semantic Indexing
LSTM Long Short Term Memory
ML Machine Learning
NER Named Entity Recognition
NLG Natural Language Generation
NLP Natural Language Processing
PBE Prediction Based Embeddings

POS Part Of Speech
RQ Research Question
RBM Restricted Boltzmann Machine
RNN Recurrent Neural Network
Seq2seq Sequence to sequence
SSAS Semantic Similarity for Abstractive Summarization
SCU Summary Content Unit
SM Schema-matching
SOM Self Organizing Map
SVD Singular Value Decomposition
TDM Term-document matrix

1 INTRODUCTION

Nowadays, Data Management, which is responsible for ensuring the security, dependability, and correctness of data, is one of the most crucial factors of every business. In essence, without proper Data Management, a company would be unable to manage, track progress, or even make decisions optimally. This concept is introduced in *Section 1.1* of the chapter. Because of this, an appropriate data analysis process aimed to investigate, cleanse, convert, and model data to find meaningful information, generate conclusions, and help decision-making is required. Data can be categorized into two types, *structured* and *unstructured data*. This difference is well highlighted in *Section 1.2*. Starting with this distinction, the Data Analysis process includes many approaches. In both cases, it is essential to manage the preliminary phase of the DPP analysis, as real-world data is frequently insufficient, inconsistent, and/or noisy. Consequently, this transformation of raw data into a comprehensible format is fundamental. In *Section 1.3*, we examine the different key elements of this process. Finally, in *Section 1.4*, we discuss our contribution whilst *Section 1.5* highlights the structure of the thesis.

1.1 INTRODUCTION TO DATA MANAGEMENT

All technologies involved in the control, access, collection, protection, distribution and analysis of data in a safe, effective and economical way fall under the category of *Data Management*. Therefore, the main objective is to assist individuals, managers and companies in making highly optimal data management and usage decisions in compliance with policies and regulations. All of this is performed with the final goal of maximizing the potential advantages of company decisions and being able to make them quickly [75].

Dealing with data management requires expertise in a challenging but essential component for the success of any business

activity in the contemporary world, where "data" is viewed as the new oil on which to focus any investment. This kind of large-scale data analysis is now possible thanks to the employment of advanced Machine Learning (ML) techniques [60].

The techniques of data manipulation and analysis have completely changed with the introduction of *Big Data* (the availability of enormous amounts of data for processing) [129]. All this is due both to the new European regulation on the protection of personal data, which has imposed a whole series of regulations on the administration and data protection and to the speed with which such data is acquired from a variety of different sources [125].

These data are extremely useful for businesses, because of their size, variety, and speed of acquisition, but they are also quite challenging to manage. As a result, the subject experts that are known as "data scientists" [1] are extremely important. These professionals manage an ever-increasing volume of data from various sources as a result of the Internet of Things (IoT) devices. So they must spend many hours cleaning up straggly databases for which the correct documentation is either lacking or for which the correct version is even unknown.

1.2 STRUCTURED AND UNSTRUCTURED DATA

Data, which might be in the form of text, photos, audio, or video, is an essential component of our daily life. These many data kinds can be broadly divided into two categories: *structured data* and *unstructured data*. The term "structured data" describes the information that has been arranged in a concise and clear manner, such as a database or spreadsheet, making it simple to search, sort, and analyze. "Unstructured data", on the other hand, is information that is not clearly organized or structured and is typically found in sources like emails, photos, or social media posts. While unstructured data presents a significant challenge for businesses because it is more difficult to manage and analyze than structured data, it should always be considered as it often offers insights that can be used to improve structured data when making business decisions.

In recent years, as businesses and organizations have worked to improve their Data Management practices, there has been much interest in integrating these two different types of data. To accomplish this, it is necessary to keep a solid platform for the collection and analysis of business data, which can be useful for a variety of tasks, including the detection, diagnosis, and correction of database system errors, the distribution of storage resources, the modification of databases, the optimization of query responses, and many others. All this is achieved through a fundamental phase of Data Management, the "Data Pre-processing (DPP)", an important step in preparing data for analysis, whose main goal is to make the data usable, accessible and meaningful, so that it can be analyzed and used to make informed decisions.

1.3 DATA PRE-PROCESSING FOR DATA ANALYSIS

Since real-world data is frequently inadequate, noisy, and/or inconsistent, the probability of collecting anomalous or false data is rather significant, given the current exponential growth of heterogeneous data sources. But accurate models and predictions may result only from high-quality data. Therefore, a process of converting the raw data into an efficient and understandable format is required. This step is known as "*Data Pre-processing*" and is a step that improves the completeness and effectiveness of datasets for data analysis for the outcomes of projects involving DM and ML techniques [14]. It speeds up knowledge discovery from datasets and may eventually impact how well ML models perform because using DPP techniques enhances the accuracy and completeness of the database. Pre-processing tasks for structured data could include adding missing values, eliminating duplicates, translating data into the appropriate formats, and normalizing data. In most cases, structured data is stored in a structured format, like a relational database, and is easily processed with the aid of structured query language like SQL or spreadsheet software. Instead, pre-processing tasks for unstructured data may include topic modeling, sentiment analysis, named entity recognition, and text cleaning. This type of data is typically stored in an unstructured format, such as text, images, or audio, and requires specialized tools and techniques for processing.

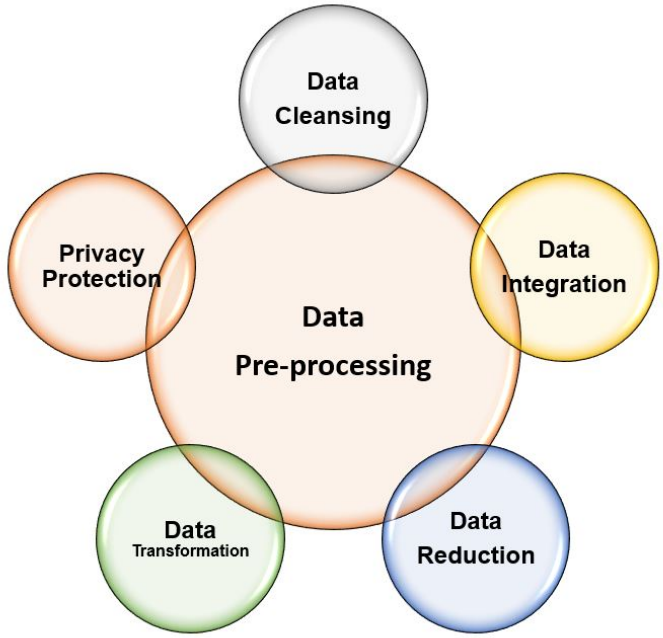


Figure 1.1: Data Pre-processing phases.

Regardless of whether the data is structured or unstructured, pre-processing should be done in a consistent and systematic manner, so that the data remains accurate and reliable throughout the analysis process. DPP is composed of fundamentally five major phases (see Figure 1.1): Data Cleansing, Data Integration, Data Reduction, Data Transformation and Privacy Protection [132].

Data Cleansing

Data Cleansing involves correcting or deleting inaccurate, damaged, improperly formatted, duplicate, or insufficient data from a dataset [80]. There are numerous ways for data to be duplicated or incorrectly categorized when merging multiple data sources. Even if results and algorithms appear to be correct, they are unreliable if the data is inaccurate.

Based on what needs to be done with the data and what the ultimate objective is to be achieved, the approaches to be employed are highly varied. There are typically various steps

to take [113]. Since not all data can be used for processing, it is helpful to start by determining the essential data fields on which to operate for a project. All pertinent information that has to be processed is then acquired and structured for this purpose. Additionally, managing null values and removing redundant data that does not provide useful information are crucial steps (through certain strategies useful for the process that must be carried out). The standardization of this cleansing procedure, so that it may be repeated on different datasets or on the same data on a regular basis is very helpful, but it is not always done.

Data Integration

Data Integration (DI) is the process of combining information from different sources into one location (such as a Data Warehouse (DW)) in order to create a unified view of the gathered data [46]. There is not a single method for completing this procedure because different companies have distinct requirements in terms of data integrity.

The fundamental concepts include data consolidation, data virtualization and data propagation.

- *Data consolidation* is the process of collecting information from several sources, eliminating redundancies, fixing errors, and aggregating the information into a single data storage [84]. Efficiency and productivity are increased when all the data is in one location. Usually, to complete this stage, a DW solution is used;

- *Data virtualization* is the technique of combining data from many sources into a virtual data layer. So, this method uses an interface to present a consistent, real-time view of data from several sources [144]. By using virtualization, it is possible to make copies (or rather views) of the data that offer an integrated perspective, whilst maintaining their original location. Companies can thereby get the benefits of an integration without having to pay the costs associated with data storage and transfer;

- *Data propagation* is the action of replicating data from one place to another with the aid of particular applications, as

the name suggests. This procedure is often event-driven and can be either synchronous or asynchronous. In essence, only a copy of the data is generated inside the locations where it will be used, and the original data will remain in all original sources.

Data Reduction

Before beginning a DM or Data Analysis process, Data Reduction is used to reduce the amount of data that is available, so as to enhance the processes in terms of convenience and efficiency. As a result, it provides a compressed view of the dataset. Despite the fact that this technique tends to decrease the amount of data, their original integrity is preserved.

This is essential when working with big data since without it, the volume of data would be unmanageable and impossible to analyze [89]. Among the main methods employed for this goal are:

- *Dimensionality reduction:* there are frequently too many factors used in ML classification challenges to make the final conclusion. These elements are essentially features, which are variables. So, using dimensionality reduction algorithms, we can reduce the number of random variables in specific situations where the majority of these features are redundant due to their correlation. Therefore, when we talk about reducing dimensionality, we imply reducing the number of these factors. It is obvious that the more features there are, the more it is necessary to increase the training dataset to create a predictive model;

- *Feature subset selection:* the selection of features involves attempting to identify a subset of the initial collection of features. This enables the use of a smaller subset to depict the issue via data modeling. Feature extraction, on the other hand, reduces data from a high dimensional space to a low dimensional space;

- *Numerosity reduction:* by selecting alternative "smaller" formats to express the data, it is also possible to reduce the

volume of data. Techniques for numerical reduction, either parametric or non-parametric, can be used for this purpose. Models created using log-linear and regression techniques are used in parametric approaches. On the other hand, non-parametric techniques retain condensed data representations using clustering, histograms and data sampling.

Data Transformation

The process of changing the format or structure of data from one format to another is known as Data Transformation. Depending on the needs, this phase can be simple or complex. As a result, a dataset can store attribute values in different units of measurement [62].

This can be accomplished using a variety of methods among which:

- *Smoothing:* algorithms can be used to filter out the noise of the dataset and learn about its core features. Additionally, smoothing makes it easier to detect even the slightest change, which aids in forecasting;
- *Aggregation:* this approach combines a dataset from multiple sources with a description of the data analysis, then saves and presents the data as a summary. The accuracy of the results depends on the quality and quantity of the data, and when both are high, the results are more helpful;
- *Discretization:* intervals are used to separate the continuous data that is displayed here. Data reduces when it is discretized;
- *Normalization:* It is a method for narrowing the representational range of the data, such as from -1 to 1 or from 0 to 1.

Privacy Protection

How to maintain the usability of data and safeguard private information from uninvited or unauthorized disclosure, is a crucial aspect of privacy protection in DM [148]. In fact, knowledge discovery from data is what is intended by the term "Data Mining" [15].

More and more often today companies exchange and publish the data they have available with other partners, for common financial and economic advantages. Nevertheless, because this data sharing frequently takes place without any kind of transformation, it causes individuals to worry about privacy violations.

Obviously, the phishing of data on the Internet, which gives access to a lot of essential personal information, has made this situation worse. The privacy of individuals has also been seriously compromised by recent developments in learning technology [72]. Because of this, it is essential to take the required precautions to protect them, both at the level of the Data Providers and in the Data Collection.

1.4 CONTRIBUTION OF THIS THESIS

The capacity to gain the most information from the data available, which can now be acquired from various sources, is one of the essential elements of Data Management. Addressing this issue is necessary for businesses to be able to organize and analyze their huge amounts of data, and there are various tools available for it. In particular, our research has focused on DI strategies and ATS algorithms.

DI involves combining data from multiple sources into a single, unified view, whilst ATS involves reducing large amounts of text to a concise and coherent summary while retaining its most important information. The relationship between these two concepts becomes particularly relevant when considering *structured* and *unstructured* data. Structured data, such as databases and spreadsheets, is well-defined and easily searchable, sortable, and analyzable. By integrating structured data from multiple sources, organizations can gain a more complete understanding of their operations. To this aim, the DI process becomes an essen-

tial tool that helps to leverage their structured data to support data-driven decision-making. On the other hand, unstructured data, such as emails, images, and social media posts, lack a clear organization or structure and can be overwhelming to process. So, ATS algorithms become an effective tool for simplifying them, as they aim to reduce it to a concise and coherent summary while retaining its most important information. In particular, the evaluation of these algorithms is essential to determine their suitability for various applications and to identify areas for improvement. In summary, the union of DI and ATS strategies can become a fundamental tool for businesses and organizations that aim to optimize their data management practices and make informed decisions based on the insights they uncover.

Starting from this premise, the first topic of this thesis focuses on Data Integration (DI). Since there are so many factors to consider for this process, including data inconsistency, query optimization, inadequate resources, scalability and implementation of support systems, the challenges that need to be addressed are continuously changing [53]. Despite the numerous approaches dedicated to DI processes, there are not many that try to fully or partially automate this process because it represents a very hard task, since DI generally requires a final step of refinement by domain specialists. For this purpose, we have dedicated our efforts to developing a semi-automatic integration process for heterogeneous databases which, whilst requiring the help of an intermediate figure who has to make some decisions in the process flow, does not require this figure to have any IT skills to complete the procedure.

The second topic addressed, instead, is related to the Automatic Text Summarization (ATS) challenge. ATS refers to the technique of automatically reducing text content whilst keeping its key points. The open question related to deciding what information is essential and what may be effectively left out makes this an open task. From this perspective, we have demonstrated in the second part of this thesis, through deep experimentation, that the evaluation metrics currently employed to confirm the accuracy of the ATS algorithms are not very efficient for the purpose for which they are used.

1.5 THESIS OUTLINE

After this Chapter 1, below we describe the contributions of each of the others.

- *Chapter 2* introduces all the ML and Deep Learning (DL) techniques used in this research work. In particular, it is focused on the description of hierarchical and non-hierarchical clustering techniques, analyzing their differences. Subsequently, neural network approaches are highlighted, making an overview of the various types considered in this study and finally, it shows the most important aspects related to IR, the process of finding relevant information from a collection of data.

- *Chapter 3* is related to the DI challenge.Following a brief presentation of the fundamental concepts of this particular topic, the State of the art is carefully explored, through the investigation of various DI methodologies. Next, our new DI approach that merges distinct heterogeneous data sources using a semi-automatic procedure is described. Using an *IR* technique, *Clustering* methods and a *Neural network*, it is possible to conclude the process involving not an IT-expert, but a figure who will only act as a link between system developers and end users, who does not need to have IT skills to complete the task. In the integrated database, the outcomes will respect all the constraints between attributes existing prior to integration.

- In *Chapter 4*, the topic of ATS algorithms is investigated. These techniques try to automatically extract important information from one or more input texts, creating summaries whilst retaining the meaning of the content. Following a brief presentation of the basic concepts related to text representation and text similarity, the State of the art of the ATS algorithms and their evaluation metrics in the literature is explored. The quality of the summaries produced by these algorithms has been evaluated using a variety of metrics, the most popular of which is ROUGE. The rigorous testing on a variety of datasets revealed that

ROUGE does not produce remarkable results because of its performance on both the EATS and AATS methods, which is similar. Furthermore, by narrowing the original reference dataset to a small field of interest, the findings are the same also considering other metrics. Moreover, a subsequent step demonstrated that multiple ATS algorithm execution generally outperforms single execution. In conclusion, it is still a long way off from creating an appropriate metric to judge the summaries created by a machine.

- Finally, *Chapter 5* highlights the conclusions of this research work, highlighting future research directions.

2 BACKGROUND

2.1 CLUSTERING METHODS

Given a large set of elements of any nature, by clustering, we try to group these elements into subsets that have features as common as possible. A Cluster can be seen as a set of elements that are similar to each other. Instead, elements in different clusters are as different as possible.

Cluster analysis, therefore, is useful to merge elements of any nature into groups of similar elements. Today we have a huge amount of data on observations of certain phenomena. So, the ability to classify these into groups of similar observations by a given feature can lead us to simplify the study of a phenomenon on a subset of information, having grouped all the observations by similarity. This will greatly simplify the work [19].

With the use of quantitative methods, cluster analysis can be used in any situation where it is necessary to collect data of any kind in a *non-intuitive* way. Finding the connections between a collection of information, and organizing it into a structure is the goal. Depending on the chosen approach, the data at hand, the research subject, multiple objectives can be achieved using the flexibility and variety of cluster analysis methodologies. The following goals can be reached using the methodologies for cluster analysis as a whole:

- identification of a real typology;
- adaptation to a model;
- group-based forecasts;
- data exploration;
- generation of research hypotheses;
- hypothesis testing;
- data reduction.

There are several subcategories of clustering approaches, one of which considers the kind of algorithm used to segment the

space. It is separated into methods for hierarchical clustering and non-hierarchical clustering.

2.1.1 *Hierarchical clustering*

The creation of a hierarchy of clusters is the goal of the clustering method known as *hierarchical clustering* where both agglomerative and divisive strategies are commonly used for it. The agglomerative hierarchical methods start from a scenario in which there are n different clusters, each of which contains a single element, and proceed to a position in which there is a single cluster that contains all n elements, by unioning progressively more nearby clusters. The hierarchical approaches of the divisive type, on the other hand, function in the opposite way.

The overall purpose of the hierarchical methodology is to obtain a sequence of partitions that can be graphically represented by means of a tree structure, known as a dendrogram (see Figure 2.1), in which the distance levels are reported on the set of ordinates and the single elements are reported on the abscisses. Each level of distance has a corresponding partition, and there are an endless number of levels of distance between any two partitions that indicate two further unions or divisions.

It is necessary to define a *measure* of cluster dissimilarity through specific metrics, that quantifies the distance between pairs of elements, and a *linkage criteria* that specifies the distance between sets of elements.

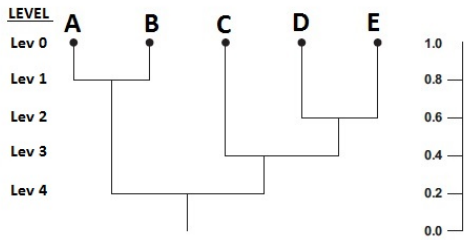


Figure 2.1: Dendogram representing hierarchical clustering.

Given two sets of elements A and B, some commonly used criteria are:

$$\textit{Complete Linkage} : max\left\{d\left(a,b\right) : a \in A, b \in B\right\} \tag{2.1}$$

$$\textit{Single Linkage} : min\left\{d\left(a,b\right) : a \in A, b \in B\right\} \tag{2.2}$$

$$\textit{Average Linkage} : \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d\left(a,b\right) \tag{2.3}$$

where d is the metric chosen to evaluate how similar two items are.

2.1.2 *Non-hierarchical clustering*

Getting a single division of the n initial elements in clusters is the aim of non-hierarchical methods. Such procedures, in contrast to hierarchical methods, permit reallocating elements that have previously been categorised to an earlier level of the analysis. The number of clusters into which to divide the entire set of n elements is defined a priori in many non-hierarchical methods of clustering, whilst in others, this number is decided during the analysis. Additionally, many of these methods call either the initial identification of a partition of the n elements in clusters or the determination of an initial set of reference points (such as an initial set of centroids, which are the places around which clusters are grouped).

Given a first partition, the non-hierarchical algorithms continue to redistribute the members of the group with the nearest centroid, until it is confirmed that for no element the distance from the centroid of a group other than that to which it belongs is smallest. The most popular technique is known as k-mean [44]. This method determines a fixed number of groups in a set of objects and needs the number of clusters to be predetermined.

The steps of this algorithm are as follows:

1. *Initialization and aggregation.* Temporary centers are chosen. Each element is inserted in the group whose centroid has a minimum distance from it;

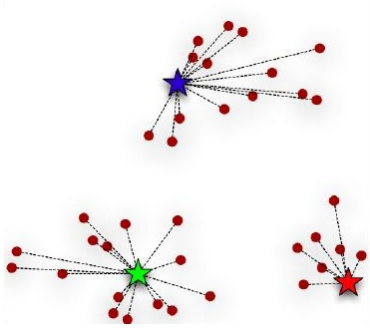


Figure 2.2: Application of the k-means algorithm.

2. *Identification of the new centroids.* The center of gravity of each obtained group is calculated going to reassign each element if the distance from it to the new center is less than that to which it was previously assigned;

3. *Identification of the new partition.* The procedure is repeated always using the minimum distance criterion;

4. *End of the procedure.* The procedure is repeated always using the minimum distance criterion.

Figure 2.2 shows the various centroids, indicated by stars coloured blue, red and green, whilst the data points are the elements that make up the clusters (the red dots near every star).

Typically, this algorithm is used when it is possible to initially identify the number of groups in which to divide the available objects.

2.2 NEURAL NETWORKS

An artificial neural network is a computational device inspired to the way that the human brain has to process information. Like a neuron, an artificial neuron has a central body and input and output terminations.

Each ramification is connected to another neuron. To trace this structure, the artificial neural network is similar to the Figure 2.3 where each connection is associated with a weight, depending on whether the neuron to which it is connected can be activated or not [16].

2.2.1 *Architecture of artificial neural networks*

The *artificial neuron* is the fundamental computing unit of the neural network and is made up of three basic elements:

1. a set of *connections* each one characterized by a weight;
2. an *adder* that adds the inputs weighted by the respective connections, producing a linear combination of the same as outputs;
3. an *activation function* to limit the amplitude of the output. In general, the amplitude of the output belongs to the range $[0, 1]$ or $[-1, 1]$.

The neuronal model also includes a threshold value which has the effect of increasing or decreasing the input to the activation function, depending on its positivity or negativity. In mathematical terms we describe a neuron k with the following equations:

$$u_k = \sum_{j=1}^{m} w_{kj} * x_j \tag{2.4}$$

$$y_k = \varphi(u_k + b_k) \tag{2.5}$$

where:

- x_i are the weights of the neuron;
- u_k is the linear combination of inputs in the neuron;

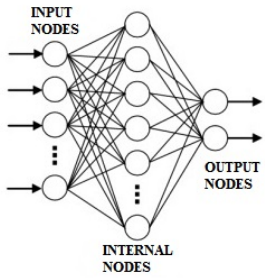


Figure 2.3: An artificial neural network model.

- b_k is the threshold value of the neuron;
- φ (x) is the activation function;
- y_k is the output generated by the neuron.

Learning and learning-rules. Neural networks can be divided into two main categories: *Supervised* and *Unsupervised*. The difference is that in the former, the network output pattern can be determined in advance. This is done by changing the weights to match certain outputs to specific inputs. On the other hand, in the latter, the learning logic is different. The output corresponding to a certain input for which the network adapts itself by some assessments of the input is not previously known.

2.2.2 *Self Organizing Maps*

Undoubtedly, the Self Organizing Map (SOM), is one of the most significant neural network topologies [114]. SOMs are part of unsupervised learning neural networks and were developed by Teuvo Kohonen between 1979 and 1982, in part based on earlier neurophysiological experiments [54].

A layer is described in the SOM as a distinct component, known as the Kohonen layer, made up of artificial neurons placed spatially in an organized manner. Theoretically, they can be thought of as one-dimensional, two-dimensional, three-dimensional, and even more than three-dimensional Kohonen layers. Two dimensions are the standard dimensionality for the Kohonen layer. This layer of Processing Elements (PE)s evolves during learning, specializing the positions of the individual PEs as indicators of the important statistical features of the input stimuli. "Feature Mapping" is another name for this procedure of spatially organizing the properties of the input data. Maps learn to classify a set of inputs by themselves, associating similar inputs to similar neurons (neighbouring neurons), therefore, the name.

The representation in a map, often two-dimensional, of the topologies connected to the input data defined in spaces with high dimensionality, is one of the most crucial applications, and as a result, it is used in data clustering issues.

2.2.3 *Recurrent Neural Networks*

A Recurrent Neural Network (RNN) is an extension of a Feed-forward Neural Network (FNN) with an internal memory more [82]. When a network is defined as *"recurrent"*, it executes the same operation for each data input, with the output produced by the current input depending on the results of the prior calculation. The created output will really be duplicated and reinserted into the RNN after it has been computed. The network, therefore, takes into account both the current input and the output it has learned from the prior input when making the final choice (see Figure 2.4). So, the primary distinction between a RNN and a traditional FNN is that the first may compute input sequences using its internal memory.

The advantages of this architecture range from the ability to process inputs of any length whilst maintaining the same model size, to the ability to account for historical information with weights shared over time. Among the disadvantages, however, are the training time, and the impossibility of considering any future input for the current state. Furthermore, the vanishing gradient issue also affects the standard RNNs that can not learn long sequences because they have a sort of short-term memory.

2.2.4 *Long Short-Term Memory networks*

Unlike traditional FNNs, LSTM [24] is an artificial neural network with feedback connections. It can process a sequence of any length by adding new information progressively into a single memory cell, with three gates determining how much new

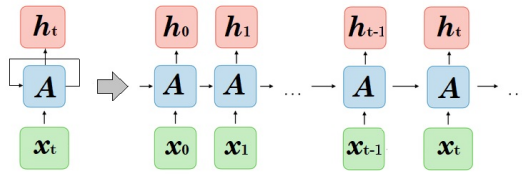


Figure 2.4: Recurrent Neural Network architecture.

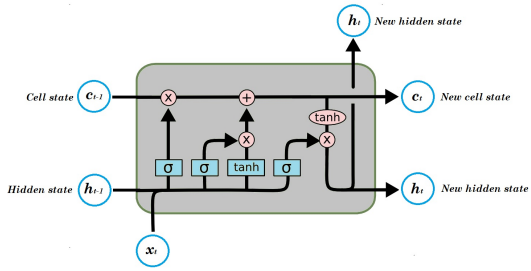


Figure 2.5: LSTM cell schema.

information should be stored, removed and shown. An auxiliary cell state that is controlled by the input, forget, and output gates constitutes a standard Long Short Term Memory (LSTM) unit (see Figure 2.5). First governs how much information enters the cell, second determines how much information is retained, and third regulates the output flow from the cell. Values from any time period are retained in the cell.

2.3 INFORMATION RETRIEVAL

Organizing, representing, storing, and providing access to the problems of the element "*information*" is the focus of IR [56, 127]. In fact, how the information is presented and organized makes it easier for users to access the information they require. However, characterizing this information is not an easy task.

An objective of the IR system, given a query, is to locate the most pertinent data for the system user in response to the query typed. In order to return the documents that are most pertinent to this topic, the IR changes its perspective on the notion that the query seeks to describe whilst attempting to interpret its semantic content.

The significant increase in the amount of data that must be kept and managed in recent years is an essential factor to emphasize because it has significantly affected the methods for document search. This required the creation of databases that could improve data management by storing and organizing the data uniformly.

Effective data organization helps the information retrieval system, but it does not satisfy the search requests that are embedded

in the text of a document. In fact, because these systems can only distinguish between documents based on their generic features rather than their content, users are left with the difficult task of manually searching through the vast amount of data to find a document that includes a specific piece of information. Additionally, an IR system could return a large number of documents in response to unselective queries, the significance of which cannot be determined before.

The user must once again go through all of the documents returned to find the one that best fits his request. As a result, it becomes necessary to arrange the documents according to their relevance. Understanding the information that the user wants to find and determining the degree of relevance of each document that is found are the two challenges in reaching this goal.

2.3.1 *Latent Semantic Indexing.*

Two primary issues are good examples of how complex IR is:

- *Synonymy*: when a meaning may be expressed by more than one word;
- *Polysemy*: when a word has more than one meaning.

When looking for such terms, one runs the risk of finding documents that have nothing to do with the topic at hand. For example, searching for the word "*net*" can lead to recovering documents relating to the Internet network rather than documents relating to a fishing net.

We would prefer to describe documents and queries through their underlying concepts, rather than through the terms that set them apart in an effort to tackle these and other difficulties of a similar nature. It should be highlighted that this hidden structure depends on the corpus (collection of documents) we are working with and is not just a straightforward many-to-many mapping between terms and concepts.

The *Latent Semantic Indexing* [47, 115] is an IR technique that attempts to capture this hidden semantic structure, through the spectral analysis of the term-document matrix A. The Singular Value Decomposition (SVD) of the term-documents matrix A

creates a new subspace into which the vectors representing the documents are projected [10].

The eigenvectors of the A^TA matrix, which correspond to the biggest eigenvalues and likely the strongest correlations between the terms, provide this subdimensional space. Queries are also projected and processed in this subdimensional space. A necessary step in implementing Latent Semantic Indexing (LSI) on a collection of n documents is the construction of the term-document matrix A (mxn), where m is the number of distinct terms present in n documents. Each document is therefore represented by an *m-dimensional* column vector.

A function of the number of times the i_{th} word appears in the j_{th} document is represented by the generic element $a_{i,j}$ of the term-documents matrix. Depending on the LSI implementation type chosen, it may be feasible to use weighting functions in such a way that each word is taken into account in both the local context of the local document and the larger context of the entire collection of documents. So, the generic element $a_{i,j}$ of A can be calculated as follows:

$$a_{i,j} = L(i,j) * G(i) \tag{2.6}$$

where $L(i,j)$ is the local weight function of term i for document j, and $G(i)$ is the global weight function for term i. There have been numerous local weight functions proposed. Following are a few of them in ascending order of efficacy:

- *Binary ($BIN_{i,j}$)*: the value of element (i,j) is equal to 1 if the term i is present in the document j, 0 otherwise;
- *Occurrences ($OCCUR_{i,j}$)*: the value of element (i,j) is equal to the number of occurrences of term i in document j;
- *Frequency ($FREQ_{i,j}$)*: the value of element (i,j) is equal to the frequency of occurrences of the term i in the document j;
- *Inverse Document Frequency ($IDF_{i,j}$)*: the value of element (i,j) is equal to the frequency of occurrences of the term i in the document j respect the total number of documents containing the term i.

Documents are represented as points (vectors) in m-dimensional space using the above-described approach. The division of these points into groups is the following step in the clustering process.

In order to determine the semantic structure of the document collection, the matrix A is next decomposed using SVD. As a result, it will be possible to map any two documents, independent of how they initially differed in the *m-dimensional* space of all distinct terms, into the same vector in the reduced space. In a sense, the set of basic vectors in the *k-dimensional* space symbolizes the range of concepts or possible interpretations for the distinct documents. In other words in this sub-space a generic text can be represented as a linear combination of the basic concepts or vectors of the space itself, so each document, word, or query is thus represented uniquely.

The search using LSI can be performed after the activities required to determine the SVD have been completed. It takes advantage of the fact that the query of the user can be expressed as a finite set of words, just like any other document, and can therefore be represented as a vector in *k-dimensional* space and be compared to all other documents. Once the method for calculating their distance has been established, it is possible to identify the documents that are conceptually closest to the query. The cosine of the angle between two vectors is an easy-to-understand indicator of closeness.

3

A SEMI-AUTOMATIC DATA INTEGRATION PROCESS OF HETEROGENEOUS DATABASES

In this chapter, we introduce a new DI methodology that does not necessitate the participation of IT-experts. In this technique, entities that are available in the sources that are syntactically/semantically related are merged by utilizing an IR approach, a clustering method and a trained neural network. We proposed some interconnections with the Company Manager, who is not required to have IT-skills and whose only impact will be to define limits and tolerance thresholds during the process, based on the interests of the business, even though the suggested approach is completely automated.

We start with an introduction about the open challenges of DI (see Section 3.1), specifically for the integration of heterogeneous databases. Section 3.2 discusses the evolution of DI over time by introducing the various current integration methodologies. Next, the State of the art is analyzed in depth (see Section 3.3), showing numerous works highlighting the different DI techniques evaluating their advantages, disadvantages and perspectives. Then we detail the proposed DI process in its various steps (Section 3.4). In Section 3.5 are provided with the experimental results on the other three systems taken into account and finally, in Section 3.6 we conclude the work outlining future perspectives.

3.1 INTRODUCTION

Companies today produce a large amount of data from their routine activities, using a variety of software tools and numerous other technologies on demand. This is because they are made up of many diversified sectors (*marketing, sales, finance, manufacturing, R&D, customer service, and so on*). Being able to extract the major marketable solutions from this data, is therefore a key challenge that today can make the difference since data is the real capital of a business [40]. This challenge is fundamental for companies

to remain competitive in the global economy in order to make quick choices.

In some circumstances, this can be a beneficial approach, but if the wrong criteria are followed, it can also lead to poor decision-making. Therefore, it really is important to take into account all the available information that, in the age of dematerialization, is replicated in digital form. Businesses have the opportunity to access these huge amounts of data and documents, from which they can undertake document analysis to determine the categories of information they represent and extract the most precious information for their purposes (from databases, social networks, geospatial data, and other sources). Document interpretation can be used to help a variety of tasks in the digital world, such as Information Retrieval (IR), document classification, and question-answering. Moreover, knowledge management can be enhanced by automatically collecting and organizing data from a sizable corpus of documents. AI offers practical strategies for improving data interaction in this scenario. It is important to be able to combine data from many sources [30], in order to give the end user a consistent representation of these data. This issue gets more challenging when more data sources are integrated because of conflicts involving duplicate data, homonyms, synonyms, and different data writing styles. DI can be performed via a number of techniques, including Extract Transform Load (ETL), Enterprise Information Integration [145], Data Federation, or Data Virtualization [133], among others. Therefore, Data Management is important for the expansion of a business and its operations. Exploring techniques that can ensure the high quality of the data is particularly necessary given that it is common for this information to come from many sources. The intrinsic integration challenges are exacerbated by poorer data quality. These sources are distributed, autonomous, and heterogeneous in addition to having various schemes and degrees of quality. As a result, the main objective of a DI system will be to deliver accurate, complete, timely, and consistent data output [18]. However, the outcomes are frequently not very trustworthy if end users and system developers do not work closely together.

The review of the state of the art highlighted that the current research proposals attempt to make the DI procedures fully or at

least partially automatic thanks to the involvement of a IT-expert because the generally produced automatic outcomes typically need to be enhanced. With the help of ML and IR techniques, we, therefore, propose in this research work a semi-automatic procedure that enables one to implement the DI process of various sources in a single reconciled data source schema without the need for IT-experts to be involved. The presence of a Company Manager is all that is required; this person is in charge of determining the acceptable boundaries and error limits for the adopted methodology. Various pre-processed input documents are subjected to both syntactic and semantic analyses as part of the DI process, which then chooses the tables that can be merged. The final results have an accuracy in the $99\% - 100\%$ range for systems with consistent input data.

3.2 BACKGROUND

3.2.1 *Data Integration Evolution*

Data management was quite easy in the early days of computers era because there were few software programs available that produced a small amount of data that was simply arranged in a single database. However, as old systems needed new capabilities added, the demand to manage more data also steadily increased over time. Thus, the first systems capable of handling numerous databases were built, and data were arranged into various containers according to the purposes for which it was being held [81].

Unfortunately, there was often data redundancy between one data container and another, and these data were also not synchronized. The necessity to reduce redundancy and have well-integrated data resulted from this context [57]. Making decisions for a company without being able to rely on integrated information posed a significant danger. Due to this requirement for integrated data, the first DI systems were developed that could modify data, although in a simple manner, to make it compatible with the target database. However, these were simple functionalities that did not allow us to avoid problems when it was necessary to make use of several simultaneous transformations.

There were many challenges to be faced, related to the number of programs to be integrated, the number of sources in continuous growth, the increase in the type of data to be managed, the heterogeneity of operating systems and DBMS on the market and last but not least the need to evaluate and improve the quality of the data [157].

Thus, from the perspective of the capabilities provided, the DI software was effective to a certain extent, but the continuous increase in data produced processing performance issues. Crucial problems about the performances that needed to be adhered to for which the various software was modified to run in parallel on multiple computers were added to the already existing ones [34].

Starting in the 1990s, structured databases and query interfaces within businesses proliferated, and the possibility for finding, exchanging, and updating structured or unstructured data increased [36]. In conclusion, we can distinguish three main phases in the evolution of DI systems:

- *First generation*: software based on programming languages;
- *Second generation*: software that supports the ETL steps and provides a Graphical User Interface (GUI) and graphical tools;
- *Third generation*: DI Suite, software packages consisting of a collection of different tools that share the same metadata base.

3.2.2 *Data Integration methodologies*

The concept of "Data Integration" refers to all the procedures and actions that enable us to combine two or more databases from various sources into a single database or view that can be easily accessed by the end user in order to conduct integrated data analysis tasks. In reality, using integrated data frequently leads to improved business outcomes.

Numerous DI apps serve as the basis of information and data infrastructures, ensuring the potential of resolving common data-sharing issues. These also enable the data to be used for all

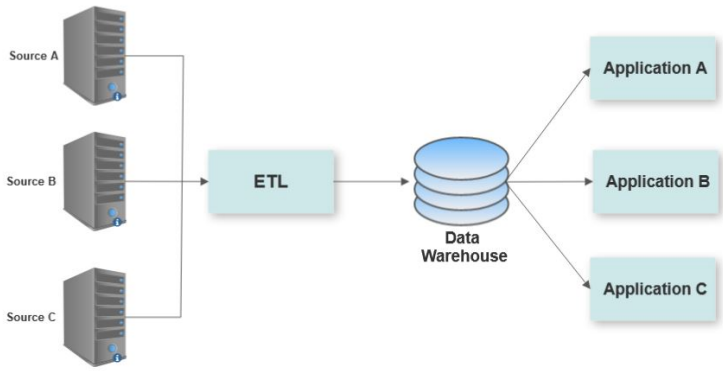


Figure 3.1: Schematic representation of a Data Consolidation solution.

company applications and business procedures. Data Consolidation, Data Propagation, and Data Federation [139] are the three basic integration techniques.

3.2.2.1 *Data Consolidation.*

By "consolidation," we refer to the procedure of collecting data from various sources inside the organization, cleaning it, and then fusing it into a single permanent and historicized database, such as a Data Warehouse, through the use of ETL methodologies [84].

Without a doubt, this strategy is the one that businesses use more often because it makes having a 360-degree view of their own organization more simpler. It also favours the development of business intelligence tools, making complex analyses and reports of company data possible. However, whether batch or real-time integration processes are used, as well as how frequently the data in the integrated database is updated, will largely determine how long it takes to develop a system of this kind. Figure 3.1 shows this type of approach.

3.2.2.2 *Data Propagation.*

While data transmission in the Data Warehouse occurs at regular intervals in a consolidated system, in a propagation-based system, data flow is constantly changing over time [51]. This strategy is popular among businesses since it is easy to adopt.

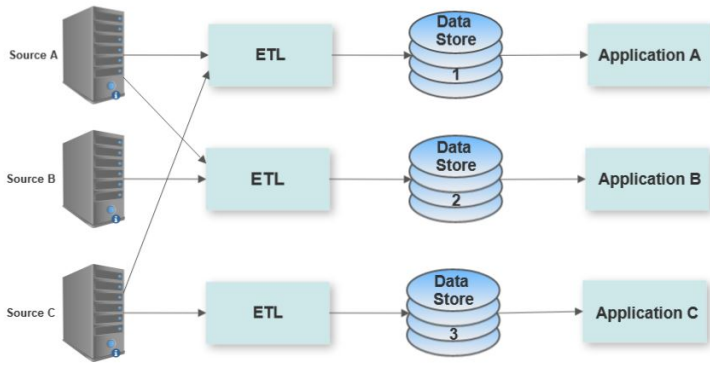


Figure 3.2: Schematic representation of a Data Propagation solution.

Specifically, using this kind of system, when an application needs to manage the data that are stored in another system, an automatic procedure is developed whose primary task is copying the appropriate data into a database that is specifically dedicated to the application in question (Figure 3.2). This includes assembling information from the operating and management systems of the organization into a specific Data Store for each application.

Although this data transfer may be bidirectional in more complex systems, in practice most systems are unidirectional due to lower implementation costs and simpler technical requirements. The challenges that arise with this technique are related to the complexity of keeping the different data containers synchronized, as the larger the system becomes, the greater the difficulty in guaranteeing the consistency of the data.

3.2.2.3 *Data Federation.*

The data model for a Data Federation solution is shown in Figure 3.3. Data Federation actually refers to the creation of a virtual platform that unifies various different, heterogeneous data sources [39].

It is necessary to use metadata or connection schemes that can link the various sources in order to guarantee data access through a single virtual view. The main benefit of this solution is

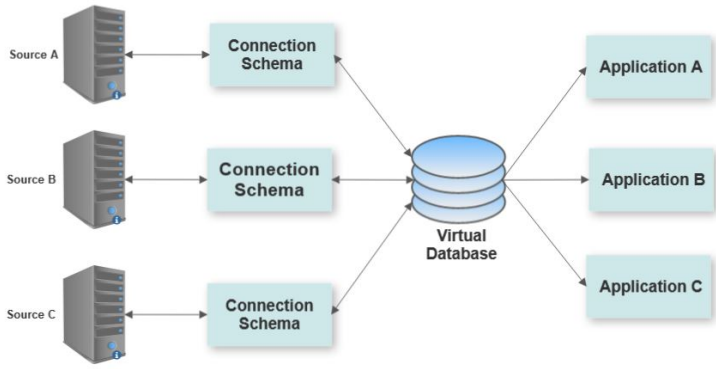


Figure 3.3: Schematic representation of a Data Federation solution.

the elimination of waiting times and a higher degree of flexibility when compared to a data consolidation solution.

This kind of strategy is typically applied to provide uniform access to several databases situated in various locations, for data synchronization, or in areas with numerous data sources. Compared to creating a data warehouse, implementing this solution proves to be a highly expensive operation, but even if it is economically disadvantageous, there are times when it is the ideal option.

3.2.3 *Schema-matching and Schema-mapping*

As stated in *Subsection 3.2.2*, Data Integration (DI) refers to the process of combining different data sources into one unified repository, and this process is used to streamline operations and make data more accessible for analysis. In order to implement a DI process, it is necessary to distinguish between Schema-matching and Schema-mapping, even if the two terms are sometimes used interchangeably. *Schema-matching (SM)* is the process of comparing two different data schemes and finding semantic correspondences between them whilst *Schema mapping* aims to somehow transform a schema element [29]. A current open challenge for the DI process is to increasingly automate these two processes since it is not possible to fully determine automatically the various correspondences between two schemes, primarily

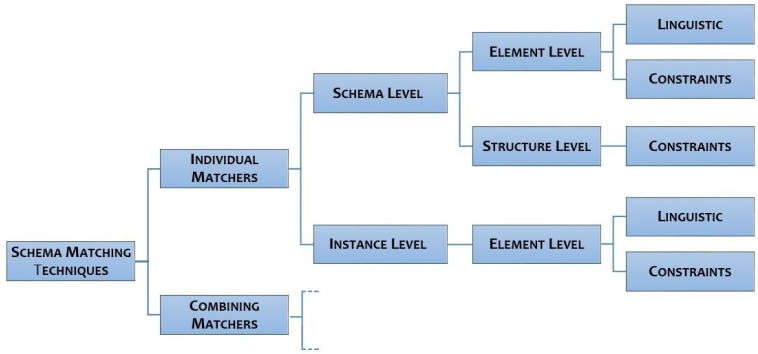


Figure 3.4: Schema-matching techniques.

because their semantics differ and are frequently not described. The main SM approaches [12, 106] are represented in Figure 3.4. The first distinction is between:

- *Individual matchers* that generate a mapping based on a single matching criteria;
- *Combining matchers* that are the result of mixing different matchers.

The further different classifications of matchers may include techniques that consider *schema* information (i.e. name, description, data type...) or only *instance* data (i.e. contents). A second classification is based on the consideration of only schema *elements* (i.e. attributes) or larger schema *structures* (i.e. groups of components). Finally, it can be considered a difference between techniques that take into account *linguistic* specifics (i.e., names and textual specifications of schema elements) or *constraints* (i.e., keys and relationships).

3.3 RELATED WORKS

The challenge of determining increasingly automatic DI procedures is still an open challenge because almost 10% of DI costs are related to the fundamental work of IT-designers. This evidence comes from an in-depth study proposed in *Papotti et al.* [99],

where it is also presented an extensible framework for the automatic effort estimation for mapping and cleaning jobs in a DI project. Since each of these tasks necessitates at least some and frequently considerable participation from the IT-experts during configuration, execution, and analysis, this assessment is an important consideration for determining the overall costs of a company. To this aim, the suggested framework provides a number of very useful indicators and methods for calculating the difficulty and overall cost of DI in many different cases, whilst accounting for heterogeneities in instances and schemes.

Finding the corresponding attributes of a source and target database schema during the Schema-matching (SM) phase is one of the most difficult aspects of integrating heterogeneous databases for the DI process. This is a significant first step towards standardizing databases and improving the modeling and application of DI techniques. The latter issue still necessitates a great deal of physical labour and has few solutions. The great majority of solutions to this problem does not consider the actual data content but instead relies solely on attribute schema information. From this perspective, *Yang et al.* [150] propose an

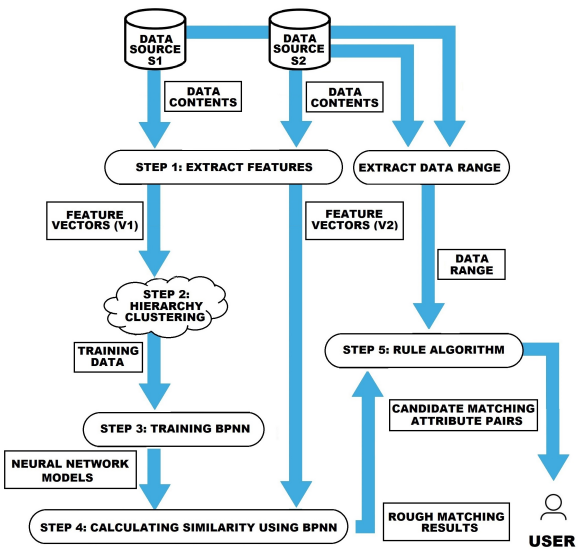


Figure 3.5: Schema-matching procedure. (*Yang et al.* [150])

integration technique of heterogeneous databases, through the *content schema-matching* process. Two parts make up this process: in the first, a set of neural networks are trained to analyze data patterns in order to generate candidate pairings for matching; in the second, a rules-based algorithm is applied to filter the previously stated set in order to find the pairs that effectively form a matching. In this case, the content of data is defined by two aspects: the *data pattern* (distribution, composition, and statistical features) and the *data range* (the set of instances or words that occur most frequently and not all instances of data attributes). The foundation of *content schema-matching* is the idea that two attributes from different data sources are similar if their data patterns or data ranges are similar. The procedure then takes two sources, S_1 and S_2, as input, extracts the features of S_1, employs clustering to identify the most important M categories (M is selected by the user), and trains a neural network to classify the data patterns of the attribute values of S1. Once the data patterns for the S_2 attributes have been extracted, the neural network is then used to compare them to the S_1 training attributes. Following this first, "raw," mapping, a rules-based approach is used to filter out ambiguous mappings and acquire the sets of attribute pairings that truly make up a matching. The described SM process is illustrated in Figure 3.5. Since no classification criteria are stated for these rules, this approach necessitates the presence of an IT-expert who can specify the rules to be implemented, making it less than fully automatic. This weight encourages to give up on this design strategy in order to create an information system that is as automatic as possible.

Bergamaschi et al. [13] propose to use the open-source DI system MOMIS (***M**ediator Envir**O**nment for **M**ultiple **I**nformation **S**ources*) to integrate data from many sources semi-automatically. It is directed at IT-designers since, although developing a large knowledge of application domains, they usually lack competence in DI techniques. In practice, not only do attribute mappings in conventional systems need to be predefined, but the results of the integration are frequently unpredictable until after it has been ended. IT-experts may use this tool to visualize the results of each stage of the process. They can also annotate lexical data useful for the attribute mappings, and preview the effects that are reached

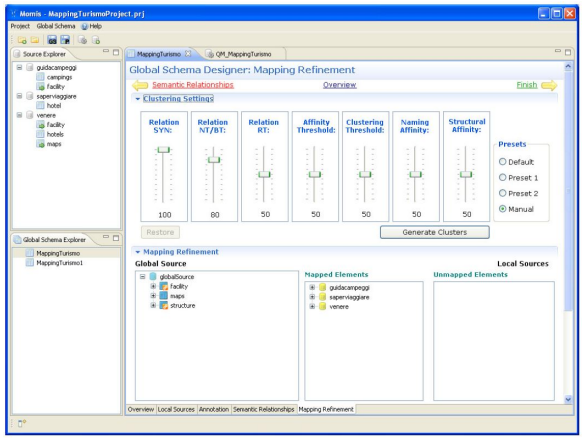


Figure 3.6: The MOMIS GUI. (*Bergamaschi et al.* [73])

as a result of the choices made. This is possible by using a variety of functionalities to fine-tune the integration results, including a top-notch graphical interface (see Figure 3.6). So, with the help of this system, it is possible to gradually improve integration process outcomes, which are normally not accessible until the procedure is finished.

Liu et al. [73] present a study that focuses on the challenge of semantic integration between different heterogeneous data sources. They examine the advantages and disadvantages of many automatic and semi-automatic SM execution methods before proposing a new general SM model that uses both data instances and source schemes. Figure 3.7 gives an overview of the integration phases. The two central phases, *Cluster schema elements* and *Schema-level matching*, in particular, both analyze

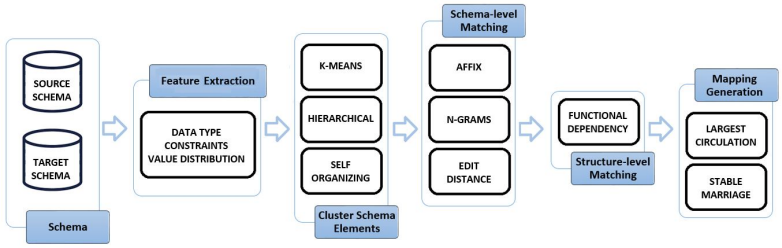


Figure 3.7: Integration phases. (*Liu et al.* [73])

the schemes of the sources to be integrated to determine which attributes are semantically related. Both hierarchical and non-hierarchical clustering algorithms (k-means) and self-organizing maps (SOM) are exploited in the clustering process. Next comes the *Structure-level Matching* phase, whose goal is to assess the similarity of attributes that are clustered in the same group whilst taking into account their context, or the data instances. In this phase, the outline provided in the previous steps can be susceptible to modification. The produced schema-mapping is used to consider the SM issue as a bipartite graph in the last phase, defined as *Mapping Generation*. This technique, however, has certain disadvantages, because determining the structure of the schema by focusing just on the source schemes and ignoring the instances, is similar to performing a syntactic or structural analysis on the attributes of the schema.

Ibrahim et al. [48] develop a three-step DI method that makes use of both structural and semantic data of the source schemes. The first step involves the users, who pay attention to the globality of the schemes, in order to decide which of them is preferable to integrate. Instead, the second step is based on five matchers relating to *Relation Schemes*, *Attribute Name*, *Data Type*, *Constraints*, and *Instance Data* to ensure the most accurate matching between the different schemes. After creating preliminary integrated schemes, the process is completed by the *Merging and Restructuring* phase. This step is necessary to assess whether the semantics of the schemes should be maintained or need to be restructured.

The study of *Mehdi et al.* [83] focuses on the issue that all instances, even those with numeric values, are processed as strings in SM instance-based techniques. This type of strategy is required because, when attribute names are compressed, only their values can be used to establish a relationship between attributes. Since it is challenging to do statistical analyses on string values, it can be easily determined a loss of match mostly between numeric attributes. The suggested algorithm has five steps (see Figure 3.8). All the instances are grouped into three categories: alphabetic, numeric or both after defining the typology of all the attributes. For each of these attributes, a sample of instances is extracted. Then, using regular expressions and google similarity for syntactic and semantic similarity respectively, the process continues

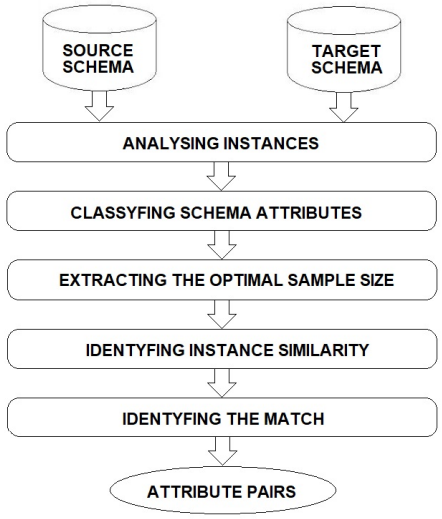


Figure 3.8: Phases of the proposed approach. (*Mehdi et al.* [83])

to determine the degree of similarity between the samples so as to identify the attributes that satisfy a predetermined similarity threshold. The findings show that this method is highly accurate in detecting *one-to-one* SM.

The *Sahay et al.* [116] research investigates a new SM method in order to successfully redesign the schemes of an integrated database. After assuming that a global dictionary of potential *one-to-many* mappings would need to be manually created, two clustering algorithms are suggested for the *one-to-one* SM method. An array of 20 features, each based on schema-specification properties and data fields, is generated to achieve this goal, one for each attribute of the source (S) and target (T) schemes. The first approach uses an unsupervised clustering model for the attributes in S to narrow the search for the corresponding attributes in T. Instead, a second approach combines each attribute from the two schemes before clustering. In this case, the distance between each T attribute and each of the S attributes in the cluster is taken into account, in order to find a match for each cluster that has at least one S and one T attribute. This method outperforms the first one in terms of performance and allows for the eventuality that a T attribute might not match any S attribute. It has been proven

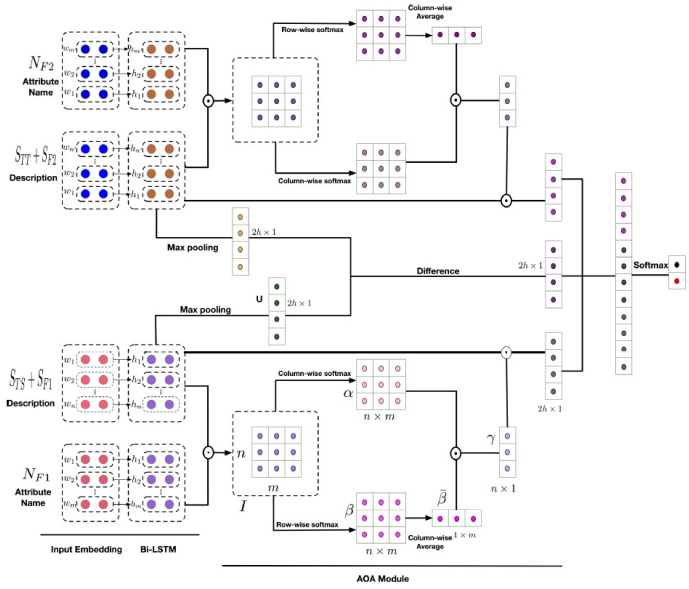


Figure 3.9: The structure of the SMAT model. (*Zhang et al.* [155])

that the Edit Distance, compared to the Euclidean Distance and the Cosine Similarity, is by far the best method for performing rigorous *one-to-one* SM.

As previously mentioned, the automation of the SM process is still mostly difficult and involves a considerable amount of manual effort. In this context, *Zhang et al.* [155] propose SMAT (***S**chema **M**atching **A**u**T**omated model*), a DL model based on Deep Neural Network (DNN) with attention, which captures the semantic correlation between source and destination scheme attributes depending on the semantic meaning of their descriptions. This system makes advantages of the most modern research in NLP, including bidirectional long short-term memory networks (BiLSTM), which may use both past and future knowledge to produce better sentence representations (see Figure 3.9). This model has the ability to automatically generate the matching between the source and destination schemes without the need to encode domain information, even if SMAT has been shown to not yet be perfectly adequate for practical application.

The basic principles underlying the generally used methodologies for identifying and generating schema mappings are linguis-

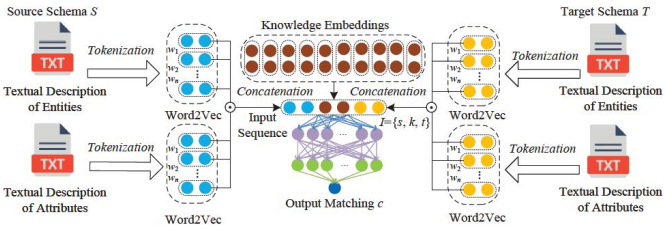


Figure 3.10: Knowledge Enriched SM Framework. (*Ma et al.* [78])

tic similarity and syntactic interpretation. These approaches only deduce syntactic attribute correlations, therefore the resulting mappings cannot fully capture the semantic connection between the source and destination schemes. A novel knowledge-enriched SM framework is proposed in *Ma et al.* [78] to overcome this issue. This architecture made up of the *Knowledge Representation Learning* and the *Schema-Matching Network* (SMN) phases, take into account various steps including schema analysis, knowledge representation learning and SM network. In order to acquire the mapping results for heterogeneous DI, external knowledge bases and pretrained embeddings are introduced into the SM network, as it is evident from Figure 3.10, so to consider the SM task as a classification task.

3.4 A SEMI-AUTOMATIC DATA INTEGRATION PROCESS

Figure 3.11 provides a detailed description of the suggested methodology and the DI procedure. As can be seen from the image, the input sources are initially converted into a suitable form necessary for the two subsequent syntactic and semantic analyses. These ones elaborate their inputs so to determine the specifics of which tables should be merged, depending on what has been processed for that particular analysis. As a result, the Company Manager will be in a great position to choose the best and most believable hypothesis for the next automatic integration of the discovered tables, using all the tools at his disposal.

Through a running example that starts with the data sources and ends with the reconciled schema, we present a step-by-step

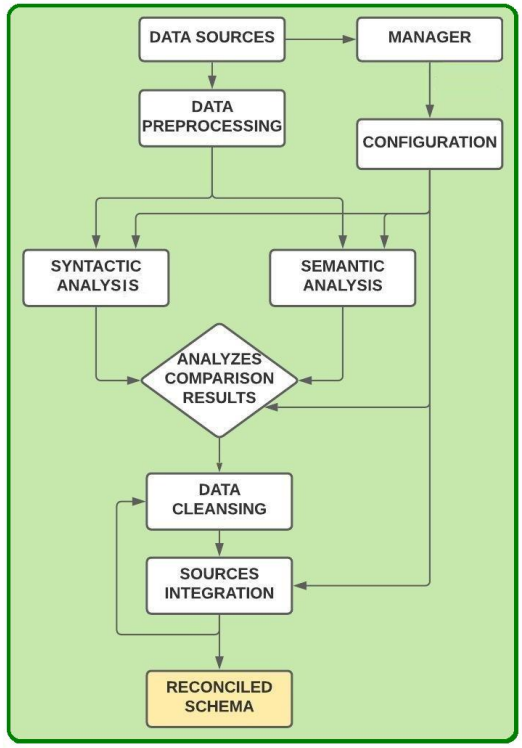


Figure 3.11: The proposed Data Integration methodology.

description of every single phase of the stated DI process to better understand the whole process.

3.4.1 *Data Sources phase*

If the data sources being integrated represent the same real-world domain, then a DI procedure makes sense. Due to this, we took into account four systems made up of companies that share these traits, when developing the technique we are about to outline. Each of them specifically represents a cohesive world where the many organizational units interact in shared processes with actors who partly or entirely share [100]. Four systems were taken into account in this research effort, one of which (related to the *Cooking system*) is currently open and available to the general

public. The other three may not be publicly available since they include private data subject to privacy protection.

For our in-depth analysis, we take into account the *CarShopping system*, which is related to a multinational business that purchases model vehicles from producers and distributes them to retailers all over the world.

The two primary relational databases used by this company are shown in Figure 3.12:

1. *Human Resources Management* consisting of the tables *clients*, *offices* and *employees*;
2. *Marketing and Sales* with different tables containing, among others,the tables *customers*, *orders* and *products*.

The data tables from these databases are extracted and stored in CSV files (*comma separated values*).

3.4.2 *Data Pre-processing phase*

When we need to analyze data and perform any classification technique, it is necessary to transform textual documents into an

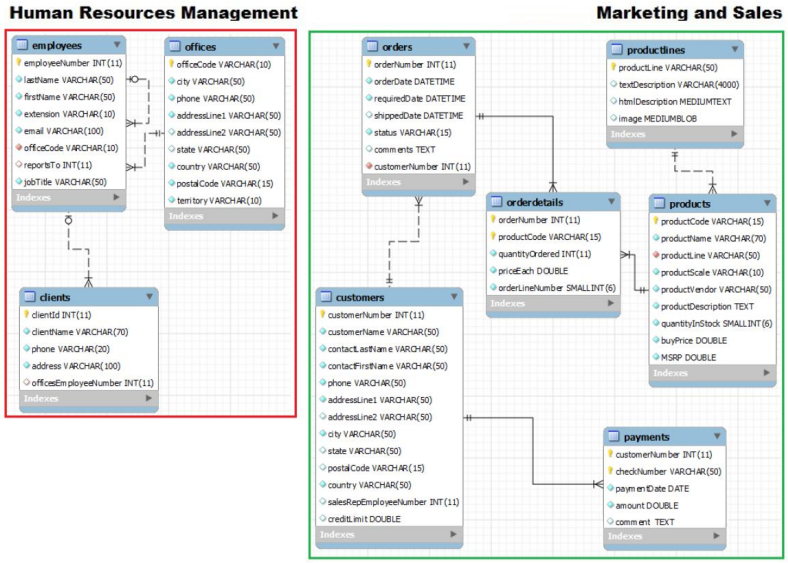


Figure 3.12: The Database CarShopping.

algebraic model in order to be able to process them [4]. An important step in identifying the shared concepts among the different tables of the data sources is the SM process, a challenging task for multiple reasons [108]. To do this, it is necessary a DPP phase to standardize and make the schemes comparable (these could differ for many features like names, structures, abbreviations or synonyms and so on). For this purpose, we adopted an IR technique [25, 56], the *Latent Semantic Indexing (LSI)* [45, 47, 115]. IR is used for two major problems: *Synonymy* (terms with similar meanings) and *Polysemy* (terms with different context meanings). These two issues can sometimes cause inconsistencies in searches, increasing the possibility of retrieving documents unrelated to the topic of interest. So, the use of LSI tries to express documents in terms of the concepts that represent them rather than their terms.

Starting with this premise, we used a DPP phase based on the *normalization* of the input documents of the text mining phase [7], which are stored in relational database tables. Tables with solely numeric data are not considered in the process because they do not provide any useful semantic information. The normalization reduces the amount of redundant information and speeds up computation. In particular, we proceed:

a) removing the columns that have a high percentage of null values (in our instance, at least 70%) from tables, as null values make the Term-document matrix (TDM) sparse and do not give enough information to find similarities between documents, increasing calculation time;

b) removing punctuation and white spaces since they are unnecessary and do not add anything to the text

c) changing each word into a lowercase word, for processing convention;

d) removing stop-words, which are terms with a very high frequency that do not offer any important information to distinguish the topics addressed in a document (e.g., articles, prepositions, conjunctions, and so on);

e) executing stemming algorithms [52], which map inflected form of words into their corresponding root form;

Table 3.1: The Term-document matrix

	$\mathbf{d_1}$	$\mathbf{d_2}$	...	$\mathbf{d_j}$	...	$\mathbf{d_n}$
$\mathbf{t_1}$	$occ_{(1,1)}$	$occ_{(1,2)}$	...	$occ_{(1,j)}$	...	$occ_{(1,n)}$
$\mathbf{t_2}$	$occ_{(2,1)}$	$occ_{(2,2)}$	...	$occ_{(2,j)}$	...	$occ_{(2,n)}$
...	...	...	...	...	...	...
$\mathbf{t_i}$	$occ_{(i,1)}$	$occ_{(i,2)}$	...	$occ_{(i,j)}$	...	$occ_{(i,n)}$
...	...	...	...	...	...	...
$\mathbf{t_m}$	$occ_{(m,1)}$	$occ_{(m,2)}$	...	$occ_{(m,j)}$	...	$occ_{(m,n)}$

f) creating the TDM, that provides the frequency distribution of each term in each document [8].

All these actions are performed to reduce the superfluous and unnecessary information to emphasize the similarity of the documents, to speed up the computation times and for a TDM less dominated by sparse terms. So, the generic element a_{ij} of the matrix A can be calculated as follows:

$$a_{ij} = L(i,j) * G(i) \tag{3.1}$$

where $L(i,j)$ is the local weighting function of the term i for document j, and $G(i)$ is the global weighting function for the term i. The construction of the TDM is the last step of the DPP phase [8]. The matrix shows the distribution of how frequently each term appears in each document. All following efforts proceed from this point.

Let $d_1, d_2, \ldots d_n$ be the n documents to be analyzed and let $t_1, t_2, \ldots t_m$ be the different m terms in the documents. The cell of the TDM $tdm[i,j]$ represents the number of occurrences of the term i in the document j, called $occ(i,j)$ (see Table 3.1).

The frequencies of the TDM are usually weighted before the matrix is further processed. The overall weight assigned to each entry in the matrix can depend on two factors:

1. the frequency of the single term in a document;
2. the frequency of the single term in the whole document collection.

In this context, we have used the weight function tf_idf (*term frequency-inverse document frequency*), which is commonly applied in IR to measure the relevance of a term for a single document with respect to all the documents. This function grows linearly with the number of times the term appears in the document, but inversely with the frequency of the term in the collection. The objective of this practice is to provide more importance to words that appear in the text but are not used consistently. It is defined by:

$$tf\_idf_{i,j} = tf_{i,j} \times idf_i \tag{3.2}$$

The first factor $tf_{i,j}$ of the function measures the importance of the *i-th* term for the *j-th* document and is expressed by:

$$tf_{i,j} = \frac{n_{i,j}}{|d_j|} \tag{3.3}$$

where $n_{i,j}$ is the number of occurrences of the term t_i in the document d_j, whilst the denominator is the cardinality (number of terms) of the document d_j, so as to avoid favouring longer documents. Instead, the second factor measures the importance of the same *i-th* term in reation to the entire collection of documents and is expressed as:

$$idf_i = log \frac{|D|}{|\{d : t_i \in d\}|} \tag{3.4}$$

that is the logarithm of the ratio between the number of documents in the collection and the number of documents that contain the term t_i. The application of this weight function allows the computing of the TDM.
The TDM for the *CarShopping* system was found to be 1,187 terms long for 8 documents.

3.4.3 *Syntactic Analysis phase*

A first possible analysis of the data is the syntactic analysis, which is based on the syntax of the data and therefore on the

information of the same. This information includes the names, the data types, as well as the domains and data descriptive information. This type of analysis is performed on the computed TDM splitting the work in two sub-phases:

- Calculation of the Levenshtein Distance (LD) matrix [156];
- Cluster analysis.

3.4.3.1 *Calculation of the LD matrix.*

The LD-matrix is computed starting from the TDM in which each document is represented as a single column vector, by applying the LD metric to all possible pairs of its vectors. By using this so computed matrix, the syntactic differences between the various documents will be compared, determining how dissimilar the documents are from one another.

3.4.3.2 *Cluster analysis.*

The LD matrix is the input for the cluster analysis. Thanks to the discovered syntactic similarity, this enables us to understand how each document is distributed among the clusters. The approach that has been implemented makes use of both *hierarchical* and *non-hierarchical* clustering [19].

1. *Hierarchical clustering* is performed with the only purpose of viewing the dendrogram, which gives a first global view of the possible distribution of documents by similarity. We

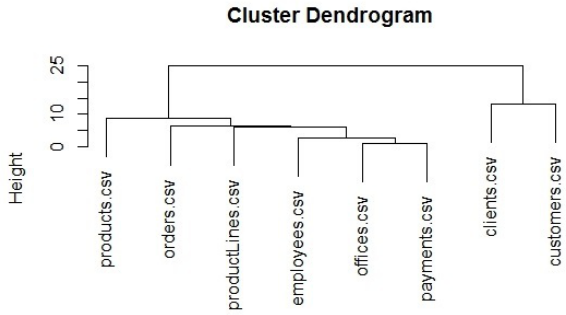


Figure 3.13: Dendrogram from *Syntactic analysis*.

used an agglomerative hierarchical clustering with the LD metric and the *complete linkage* criterion [109]. In Figure 3.13 is shown the dendrogram for the *CarShopping* System. It is visible in it the affinity between the tables *clients* and *customers*.

2. For *non-hierarchical clustering* it was used the *k-means* algorithm [68] by setting its parameters as follows:
   - *number of clusters* = half of the number of input documents;
   - *number of iterations* = 10.000;

   and using the LD matrix as metric. To ensure the stability of the algorithm before it is run, the following requirements have been established:
   a) Run the k-means algorithm until there are no more advances in the output of the *Between Sum of Squares / Total Sum of Squares (BSS/TSS)* index. Based on our testing, the k-means algorithm must be run at least five times in succession. The BSS/TSS index denotes the goodness of the clustering [21]. Hopefully, we want a clustering that has the properties of internal cohesion and external separation (i.e. the BSS/TSS ratio as close to 1 as possible).
   b) Each time the algorithm runs, the cluster holding a single document must be eliminated from the findings; the new group of documents, with the threshold now being lowered by one, will serve as the input for the subsequent iteration.
   c) The minimum number of possible clusters is 2.

Table 3.2: K-Means results for *Syntactic analysis*

Iter	Param		Documents	Clusters	Bss/Tss	Doc Out
	k	N				
1	4	8	1,2,3,4,5,6,7,8	C_1={1,2}, C_2={3,4,6}, C_3={5,7}, C_4={8}	85,0%	8
2	3	7	1,2,3,4,5,6,7	C_1={1,2}, C_2={3,4,6}, C_3={5,7}	82,5%	

1=clients 2=customers 3=employees 4=offices 5=orders 6=payments 7=productlines 8=products

The selection of these constraints, which was influenced by a number of tests as well as certain heuristics from the literature [104], considerably affects the outcome of the DI process. The findings are shown in Table 3.2, where each row represents a k-means algorithm iteration output. In this table:

- (*Iter*) indicates the current iteration number;
- (k and N) are respectively the number of clusters and of documents in the current iteration;
- (*Documents*) reveals the documents that are still present since one document is removed after each iteration and may be seen in the *(Doc Out)* column;
- (*Clusters*) shows the clusters that formed in the specific algorithm iteration;
- (*Bss/Tss*) is the k-means cohesion coefficient for that iteration.

Based on the table, two algorithm iterations are necessary to cluster the initial seven documents into three clusters, with a fitting rate of 82.5%. Clusters C_1 and C_2, which were discovered during the second execution, point to the possibility of matching between the two source schemes.

3.4.4 *Semantic Analysis phase*

Semantic analysis [149] is the second type of investigation that may be performed on the input files that have already been processed, with the aim of discovering their semantic correlations. The procedure begins with the tables from the two normalized schemes, then using the TDM doing the following three fundamental steps:

1. Unsupervised training of the SOM network [54, 114];
2. Calculation of the *covariance matrix* of the weights assigned by the SOM to the documents;
3. Cluster analysis.

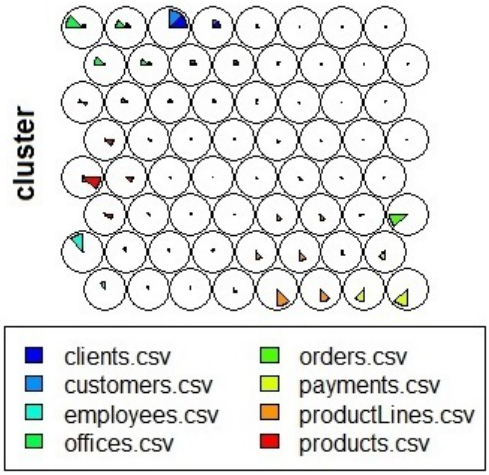


Figure 3.14: SOM grid outcomes (64 Unit SOM).

3.4.4.1 *Unsupervised training of the SOM network.*

The training set for SOM is made up of the reduced TDM, obtained by removing the scattered terms. As regards the *CarShopping* system under investigation, the sparsity ratio was set at 0.95 because values lower than this threshold would have removed too many terms (more than half of the total) whilst higher values would have resulted in the removal of too few terms (about 1%). This sparsity ratio setting led to the definitive elimination of 15% of the total words. The output of this training is visible on a two-dimensional grid, where each SOM unit (or cell) contains similar documents. Specifically, neighbouring cells will contain groups of documents that share similar features. Figure 3.14 shows the result obtained for the *CarShopping* system.

3.4.4.2 *Calculation of the* covariance matrix *of the weights assigned by the SOM to the documents.*

Following the initial training of the SOM network, the semantic analysis continues by performing an unsupervised classification of the output of this network, i.e. the matrix of the output weights assigned to the various documents. Considering n input documents, the weights of the output vectors SOM will be contained in a matrix of size $n^2 \times n$, in which the contents of each unit SOM

Table 3.3: Approximated *covariance matrix*

	1	2	3	4	5	6	7	8
1	0.02080	0.02043	-0.002042	0.00387	-0.001738	-0.002098	-0.00429	-0.001707
2	0.02043	0.02089	-0.002203	0.00386	-0.002098	-0.002385	-0.00525	-0.002378
3	-0.00204	-0.00220	0.018745	-0.00152	-0.000555	-0.001348	-0.00283	-0.000836
4	0.00387	0.00386	-0.001528	0.03049	-0.001591	-0.001802	-0.00471	-0.001724
5	-0.00173	-0.00209	-0.000555	-0.00159	0.016592	0.000522	-0.00115	-0.000667
6	-0.00209	-0.00238	-0.001348	-0.00180	0.000522	0.026787	-0.00205	-0.001862
7	-0.00429	-0.00525	-0.002833	-0.00471	-0.001153	-0.002056	0.03578	0.002031
8	-0.00170	-0.00237	-0.000836	-0.00172	-0.000667	-0.001862	0.00203	0.024584

1=clients 2=customers 3=employees 4=offices 5=orders 6=payments 7=productlines 8=products

will be saved. This array will be normalized to the range $[0, 1]$. It is important to emphasize that in this phase, all insignificant elements will be excluded from the processing. These elements will be the ones that will have values significantly lower than the maximum weight given to the SOM units because they are not competitive. To this aim, we set them to 0 in the matrix. As a result, each column vector will have n^2 components (the *i-th* component will represent the weight assigned by SOM for that vector to the *i-th* SOM unit), and the covariance will then be determined for each pair of column vectors. The outcome is referred to as *covariance matrix*. It connects every possible pair of documents and shows how each variable changes in relation to the others. The subsequent stage of cluster analysis will use this matrix as its input. The *covariance matrix* found for the *CarShopping* system is shown in Table 3.3.

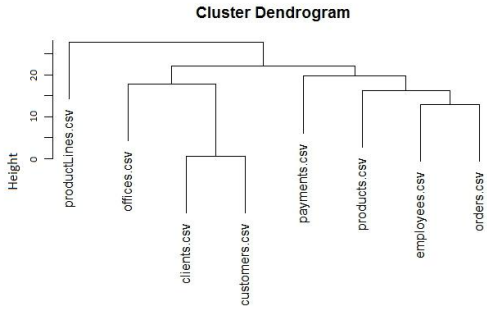


Figure 3.15: Dendrogram from *Semantic analysis*.

3.4.4.3 *Cluster analysis.*

The clustering phase, which concludes the semantic investigation, aims to comprehend how documents are distributed across the different clusters by using the covariance matrix to identify semantic similarities. The procedure will take the same direction as the syntactic analysis. The dendrogram in Figure 3.15 shows the outcomes of the hierarchical clustering for the *CarShopping* system, and it is evident that the *clients* and *customers* tables have a great affinity. Table 3.4 instead, displays the final k-means results.

3.4.5 *Comparison of syntactic and semantic analyses findings*

The final step of the integration process involves comparing the findings of the syntactic and semantic analyses of data, in order to determine which of them best *"captures"* the correspondence between the data sources. With the help of a common approach of clustering, the two analyses reach their respective conclusions.

These results, however, cannot be directly compared numerically because the starting inputs are different. Therefore, in order to somehow relate what was obtained in the two cases, it is first necessary to identify an index that can numerically represent the goodness of clustering. Since k-means is used as the clustering algorithm in both analyses, the one and only parameter that can serve as an *"index of goodness"* for this purpose, is the one already defined for the output of this algorithm: the BSS/TSS. This index is in fact very representative of the k-means algorithm because it indicates the goodness of the input data fitting.

Table 3.4: K-Means results for *Semantic analysis*

Iter	Param		Documents	Clusters	Bss/Tss	Doc Out
	k	N				
1	4	8	1,2,3,4,5,6,7,8	C_1={1,2}, C_2={3,5,6}, C_3={7,8}, C_4={4}	86,4%	4
2	3	7	1,2,3,4,5,6,7	C_1={1,2}, C_2={3,5,6}, C_3={7,8}	82,1%	

1=clients 2=customers 3=employees 4=offices 5=orders 6=payments 7=productlines 8=products

Since this index cannot be directly compared numerically for the two analyses for the reasons stated above, it is necessary to define a new index, I_{SynSem}, which considers both the syntactic and semantic analysis results. The new index will be defined as the linear combination of the k-means fitting indexes of the two analyses:

$$I_{SynSem} = \alpha * Max_{syn}(Bss/Tss) + \beta * Max_{sem}(Bss/Tss) \quad (3.5)$$

where :

- $\alpha, \beta \geq 0$;
- $\alpha + \beta = 1$;
- Max_{syn} is the Best Index for the syntactic analysis;
- Max_{sem} is the Best Index for the semantic analysis.

By examining the dendrograms, the SOM map, the results of the two k-means algorithms, as well as his business skills, the Company Manager will be able to determine which of the two analyses provided the best clustering findings. According to all these tools at his disposal, he can set *alpha* and *beta* parameters. Semantic analysis is often preferred because it is more exhaustive. Because of this, the Company Manager usually chooses $\beta > \alpha$.

The $\{clients, customers\}$ files form a cluster that is highlighted by the intersection of syntactic (82.5% fitting) and semantic (82.1%

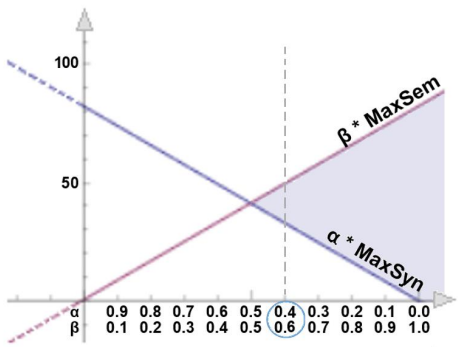


Figure 3.16: Choice of α and β.

fitting) analyses. As a result, the integration is focused on these two tables because they both describe the same domain of interest. Since the semantic analysis is more thorough, $\alpha = 0.40$ and $\beta = 0.60$ could be a good choice to ensure that it prevails over the syntactic analysis (see Figure 3.16).

3.4.6 *Data Cleansing phase*

The objectives of the *Data Cleansing* phase are to discover incomplete, irrelevant or incorrect parts of the data records as well as to replace or modify the data records themselves, eliminating the dirty or coarse ones [107]. After this "cleansing", a dataset should be consistent with other similar datasets.

Several methods have been suggested to achieve this goal:

1. *Schema-Level approach:* this requires the adjustment of diagrams, to obtain the reconciled schema of the sources (for example by renaming some corresponding fields of two tables).

2. *Instance-Level approach:* this tries to have some similarity metrics between the records of different tables [154].

In this procedure, it is used the *Schema-Level* approach [124], which attempts to establish correspondence between different files or database structures by taking advantage of appropriately defined similarities. We identify similar tables by clusters and discover the fields that may correspond using the k-means method findings. Its purpose is to establish the correlation between the various tables that will be merged (in a cluster there are at least two tables to be merged).

All tables are taken into account during this phase, excluding the use of columns with only numeric values. These columns will be used afterwards to ensure the union with the other tables not considered in the integration process. As a result, the objective of the data cleansing phase is to establish a correspondence matrix between the columns of the two tables that will be joined. Although only the case of two tables is shown here, the procedure may be used for any number of tables by performing repeated iterations. Assume that there are two mergeable tables in the

Table 3.5: Match matrix for Tables *clients* and *customers*

		Customers					
		customerNumber	**customerName**	**contactLastName**	**contactFirstName**	**phone**	⋮
Clients	**clientId**	0	0	0	0	0	...
	clientName	0	**122**	0	0	0	...
	clientPhone	0	0	0	0	**121**	...
	clientAddress	0	0	0	0	0	...
	...	...	...	...	...	...	...

cluster: A (with m columns) and B (with n columns). Then we construct a $m \times n$ matrix where each cell (i, j) contains the number of elements that occur both in the *i-th* attribute of table A and e in the *j-th* attribute of table B. Only alphanumeric elements are taken into account in the comparison. In this matrix, finding the highest value for each row (or column) will reveal the attributes that need to be unified.

Table 3.5 shows part of the results for the tables *clients* and *customers* of the *CarShopping* System. We can note correspondences between the attributes *(clientName, customerName)*, and between *(clientPhone, phone)*. The two tables will then be merged using these matching attributes, which will be considered as key attributes.

3.4.7 *Sources Integration phase*

After the Data Cleansing phase, the two tables can be merged, using the join operation of the relational databases on the columns of the key attributes. Assuming A and B as the two tables to be merged, one can distinguish between:

- *Inner join:* it compares each row of table A with each row of table B, following the definition of a comparison rule. If the rule is valid for these rows, they will be combined to form a single row in the integrated table which will respectively contain all the values found in columns of A and B.

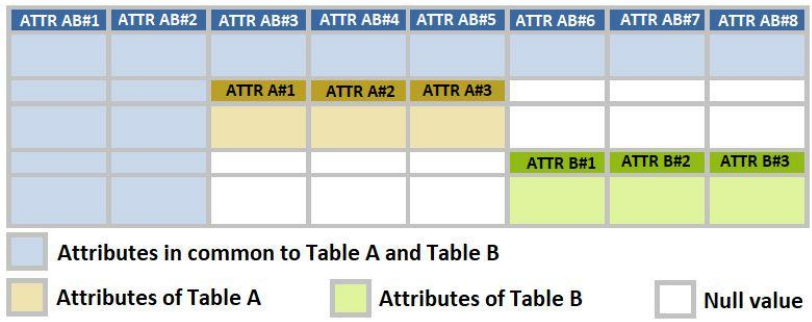


Figure 3.17: Merging of tables A and B.

- *Outer join:* here it is not required a perfect matching between the rows of tables A and B. In this case, the resulting integrated table will hold all not only the data records of both tables that match the comparison rule, but also all the other tuples of one or the other table (by one of *left outer join*, *right outer join* or *full outer join*).

In the algorithm, both inner join and full outer join were applied and the keys involved are those generated during the *data cleansing* phase. Referring to Figure 3.17, if A and B are the tables to be merged, the integrated table will first contain the columns that belong to both tables A and B (the part in blue in Figure 3.17), then the remaining columns from the two tables that do not contribute to the merging (for Table A the part in yellow, and for Table B the part in green in Figure 3.17). When there are more than two tables to be integrated, both the Sources Integration Phase and the Data Cleansing Phase can be iterated (see Figure 3.11).

For the *CarShopping* system, the integration proceeds by performing the inner join of *clients* (made up of 5 columns) and *customers* (made up of 13 columns). The new integrated table will consist of 16 columns of which 2 (*di_cust_Name, di_cust_Phone*) are shared by the input tables and map the correspondence between the attributes (*clientName, clientPhone*) of the first, with (*customerName, phone*) of the second. After this, creating the reconciled sources schema is straightforward (see Figure 3.18).

In order to ensure that the new integrated table (A+B) has all the previous connections (i.e., relationships) with those already

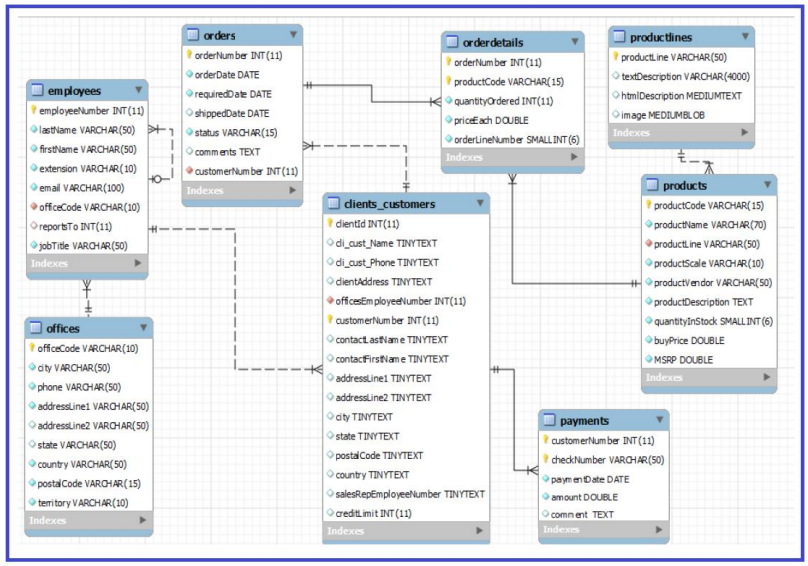


Figure 3.18: Reconciled Scheme for the *CarShopping* system.

present in the sources, which are related to the single tables of the two initial databases, it is important to emphasize that data cleansing and merging of the tables will also take into account the numeric columns because they can be foreign keys and must, therefore, be preserved.

3.5 CASE STUDY

The efficiency of the proposed methodology was evaluated by conducting an in-depth investigation of three different systems: *Panda*, *Plants* and *Cooking*. Below is a discussion of the findings.

3.5.1 *The Panda system*

System description. The PANDA system (Presences And Notes Data Analysis using Data Warehousing) is a DW developed with the aim of simplifying the analysis of university activities, with particular attention to material produced by students.

Sources. Two relational databases are used to represent the system (see Figure 3.19). Tables with only numeric attributes will

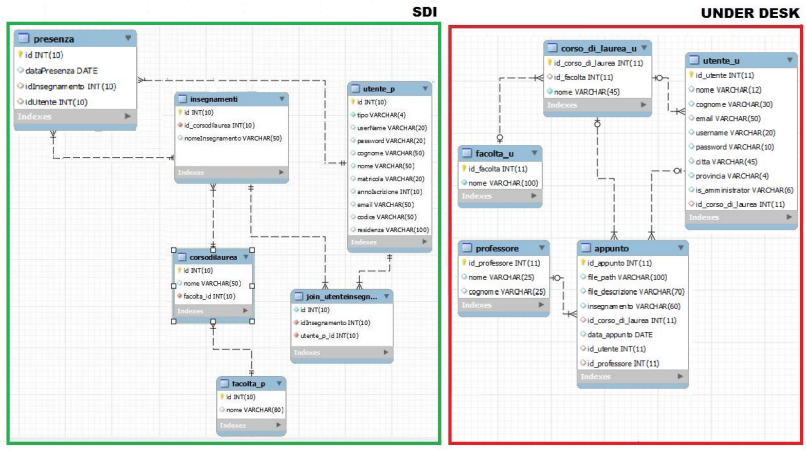


Figure 3.19: E/R diagram of data sources for the Panda System.

not be considered in the discussion, because they do not add any useful semantic information to the process.

1. *SDI (Student Digital Identification)*: it is a system for determining attendance in a university setting, that was employed by the University of Molise from 2008 to 2010. The system contains four participating tables (*corsodilaurea, facolta_p, insegnamenti, utente_p*), which stand for (*degree-ofcourse, faculty_p, teachings, user_p*), respectively. Around 18,000 tuples of attendance data and over 54,000 tuples of user information are saved in the system database;

2. *UnderDesk*: it is an online application that lets users exchange notes. The system contains five participating tables (*appunto, facolta_u, corso_di_laurea_u, professore, utente_u*) that stand for (*note, faculty_u, degree_course_u, teacher, user_u*) and can hold more than 54,000 users and 3,000 notes.

Initial considerations. The terms-documents matrix is built during the *DPP* phase. For this system, size is 42,228 terms for 9 documents. From an initial overview, it can be noted an affinity in names among the couples of tables (*facolta_p, facolta_u*), (*utente_p, utente_u*) and (*corsodilaurea, corso_di_laurea_u*).

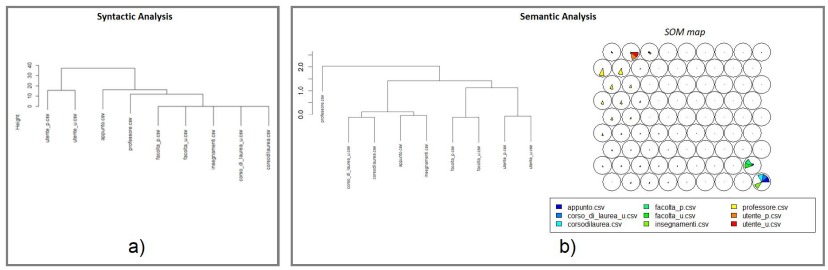


Figure 3.20: Dendrograms and SOM map results for the Panda System.

Syntactic analysis. The result of the hierarchical clustering of the syntactic analysis suggests the existence of the cluster (*utente_p*, *utente_u*) (see Figure 3.20-a). Instead, the subsequent k-means algorithm (see Table 3.6) highlights three clusters with a final fitting of 91.9%. The first puts (*utente_u*, *utente_p*) together, but is visible also a cluster for the tables (*facolta_p*, *facolta_u*) and a third cluster composed of (*corsodilaurea*, *corso_di_laurea_u*, *insegnamenti*) tables. Surely, by performing the semantic analysis, more information will become evident.

Semantic analysis. Hierarchical clustering is done in this phase too. In this case, the correlations between (*utente_p*, *utente_u*), (*facolta_p*, *facolta_u*) and (*corsodilaurea*, *corso_di_laurea_u*) are clearly shown from the dendrogram, whilst at the same time, the train-

Table 3.6: K-Means results for Panda System

Iter	Param k	Param N	Documents	Clusters	Bss/Tss	Doc Out
			K-Means for Syntactic Analysis			
1	5	9	1,2,3,4,5,6,7,8,9	C_1={1}, C_2={8,9}, C_3={2,3,6}, C_4={7}, C_5={4,5}	96,6%	7
2	4	8	1,2,3,4,5,6,8,9	C_1={1}, C_2={8,9}, C_3={2,3,6}, C_4={4,5}	96,5%	1
3	3	7	2,3,4,5,6,8,9	C_1={8,9}, C_2={2,3,6}, C_3={4,5}	91,9%	
			K-Means for Semantic Analysis			
1	5	9	1,2,3,4,5,6,7,8,9	C_1={1,6}, C_2={8,9}, C_3={2,3}, C_4={7}, C_5={4,5}	99,9%	7
2	4	8	1,2,3,4,5,6,8,9	C_1={1}, C_2={6}, C_3={2,3}, C_4={4,5,8,9}	97,3%	1
3	3	7	2,3,4,5,6,8,9	C_1={8,9}, C_2={2,3,6}, C_3={4,5}	95,2%	

1=appunto 2=corso_di_laurea_u 3=corsodilaurea 4=facolta_p 5=facolta_u
6=insegnamenti 7=professore 8=utente_p 9=utente_u

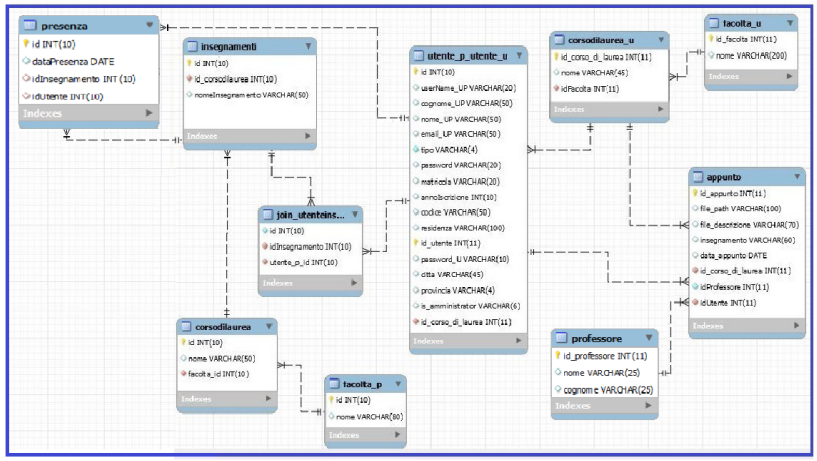


Figure 3.21: Reconciled schema for the Panda System.

ing of the SOM network reveals clusters where (*utente_p*, *utente_u*) are close by or in the same SOM unit (see Figure 3.20-b). The k-means produces the same results of the syntactic analysis (see Table 3.6) with a final fitting of 95.2%. Comparing the two dendrograms obtained reveals more information, particularly in the one produced using the semantic analysis, where four clusters are quite distinct and each contains two tables from two separate sources.

Analyses comparison. When comparing the two analyses, it can be observed that two k-means methods result in the same tables being highlighted in the different clusters. Considering the viewing of the dendrograms and outcomes from the training of the SOM network, the dendrogram of the semantic analysis distinguishes better the groupings. So, the semantic analysis prevails over the syntactic one and to this aim a good choice of the parameters could be $\alpha = 0.45$ and $\beta = 0.55$.

Final integration. Given that the comparison of the analyzes made the semantic analysis prevail, in order to verify the accuracy of the process discussed, we merge the tables of cluster C_1 (*utente_u*, *utente_p*) which contain over 54,000 tuples each. This choice is due to the consideration that the tables of cluster C_2 require two consecutive merges and those of cluster C_3 are simple

to merge. In this case, no further *data cleansing* step is required, so the merging leads to an integrated table of 17 columns, where 4 (*username*, *cognome*, *nome*, *email*) are shared by the two tables (these have 11 and 10 columns respectively). Figure 3.21 shows the reconciled schema.

3.5.2 *The Plants system*

System description. The Plants System focuses on merging two data sources from two separate flower-related companies to create a DW for the management of floral sales services.

Sources. Two relational databases are used to represent the system (see Figure 3.22). Tables with only numeric attributes will not be considered in the discussion, because they do not add any useful semantic information to the process.

1. *DB Vivaio E-commerce*: it is a database for a flower sales company that sells plants and offers several related services; its six participating tables (*azienda*, *attivita_esterna*, *cliente_vivaio*, *dipendente*, *privato*, *specie_pianta*) stand for (*company*, *extern_activity*, *flower_customer*, *employee*, *private*, *species_plant*) respectively;

2. *DB Vivaio Interventi*: it is a database for a company that only offers flower sales services and invests in activities similar to the first company; its three participating tables

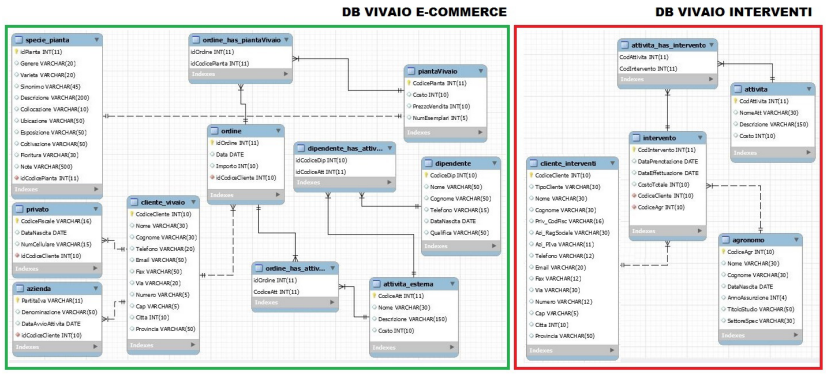


Figure 3.22: E/R diagram of data sources for the Plants System.

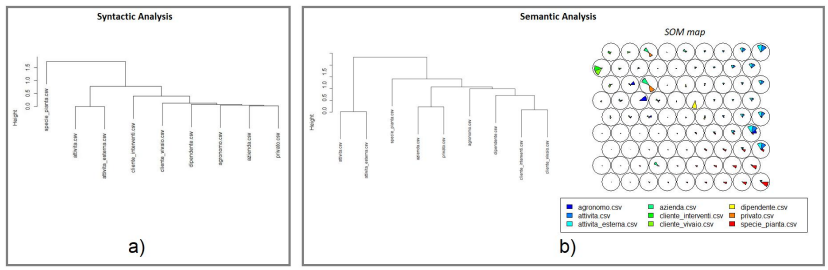


Figure 3.23: Dendrograms and SOM map results for the Plants System.

(*agronomo, attivita, cliente_interventi*) stand for (*agronomist, activities, customer_works*) respectively.

Initial considerations. The terms-documents matrix is built during the DPP phase. For this system, size is 1,323 terms for 9 documents. From an initial overview, it can be noted an affinity in names among the couples of tables (*attivita_esterna, attivita*), (*cliente_vivaio, cliente_interventi*).

Syntactic analysis. The result of the hierarchical clustering of the syntactic analysis suggests the existence of the clusters (*attivita_esterna, attivita*) and (*azienda, privato*) (see Figure 3.23-a). The (*attivita_esterna, attivita*) cluster is also highlighted from the subsequent k-means algorithm (see Table 3.7), with a final fitting of 88.7%. Therefore, it appears from this analysis that the correlation identified by taking into account the names of the tables (*cliente_vivaio, cliente_interventi*) is lost.

Semantic analysis. In this case, three clusters (*attivita_esterna, attivita*), (*azienda, privato*) and (*cliente_vivaio, cliente_interventi*) are clearly shown from the dendrogram of the hierarchical clustering. The same relationships are also evident in the SOM training results, which show three SOM units with similar document correlations (see Figure 3.23-b). Intersecting these evidences with the result of the k-means (see Table 3.7) that shows three clusters with a final fitting of 90.2%, the (*attivita_esterna, attivita*) and (*cliente_vivaio, cliente_interventi*) groups are the suggested correlations of this analysis.

Table 3.7: K-Means results for Plants System

Iter	Param k	Param N	Documents	Clusters	Bss/Tss	Doc Out
K-Means for Syntactic Analysis						
1	5	9	1,2,3,4,5,6,7,8,9	C_1={1,4,7,8}, C_2={9}, C_3={2,3}, C_4={6}, C_5={5}	99,8%	9
2	4	8	1,2,3,4,5,6,7,8	C_1={1,4,7,8}, C_2={2,3}, C_3={6}, C_4={5}	99,7%	5
3	3	7	1,2,3,4,6,7,8	C_1={1,4,7,8}, C_2={2,3}, C_3={6}	99,4%	6
4	2	6	1,2,3,4,7,8	C_1={1,4,7,8}, C_2={2,3}	88,7%	
K-Means for Semantic Analysis						
1	5	9	1,2,3,4,5,6,7,8,9	C_1={1}, C_2={2,3}, C_3={4,7,8}, C_4={9}, C_5={5,6}	92,5%	9
2	4	8	1,2,3,4,5,6,7,8	C_1={1}, C_2={2,3}, C_3={4,7,8}, C_4={5,6}	91,5%	1
3	3	7	2,3,4,5,6,7,8	C_1={2,3}, C_2={4,7,8}, C_3={5,6}	90,2%	

1=agronomo 2=attivita 3=attivitaesterna 4=azienda 5=cliente_interventi
6=cliente_vivaio 7=dipendente 8=privato 9=specie_pianta

Analyses comparison. The outcomes of the two analyses are very dissimilar. As can be visible, the documents *cliente_vivaio* and *cliente_interventi*, whose names share linguistic similarities, are omitted from the syntactic analysis outcomes. In addition, even the dendrogram of the syntactic analysis is less clear than the one of the semantic analysis. Considering all of these factors, it emerges that the semantic analysis performs better in this case as well. So, a good choice of the parameters could be $\alpha = 0.45$ and $\beta = 0.55$.

Final integration. Given that the comparison of the analyses resulted in the semantic analysis prevailing, in order to verify the accuracy of the process discussed, we merge the tables of cluster C_3 (*cliente_vivaio, cliente_interventi*). This choice is due to the consideration that the tables of cluster C_1 are simple to merge and those of cluster C_2 require two consecutive merges (being composed of three tables). In this case, no further *data cleansing* step is required, so the merging leads to an integrated table of 18 columns where 8 (*nome, cognome, telefono, email, fax, via, citta, provincia*) are shared by the two tables (these have 15 and 11 columns, respectively). Figure 3.24 shows the reconciled schema.

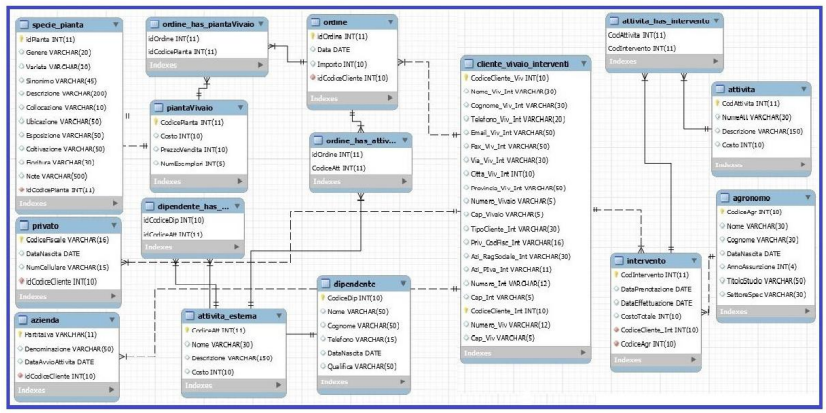


Figure 3.24: Reconciled schema for the Plants System.

3.5.3 *The Cooking system*

System description. The cooking system is a system related to the cooking industry. The starting sources are accessible via the open data catalogue, made available by the Italian Ministry of Public Administration[1].

1 http://www.dati.gov.it

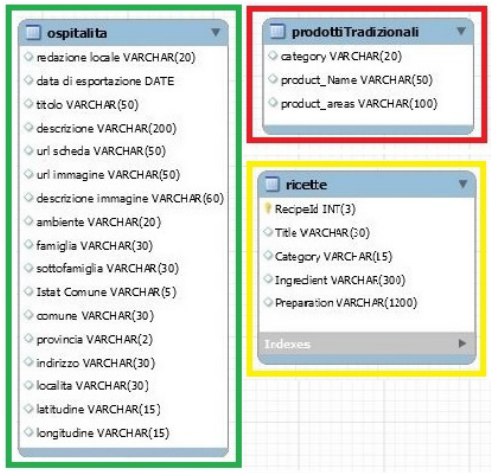


Figure 3.25: E/R diagram of data sources for the Cooking System.

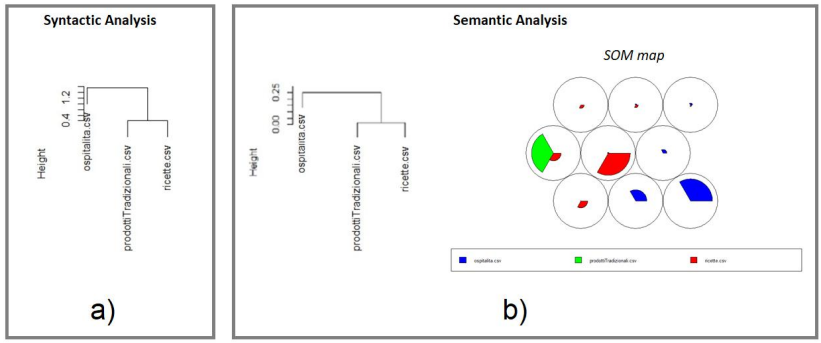


Figure 3.26: Dendrograms and SOM map results for the Cooking System.

Sources. Three *.csv* files are used to represent the system (see Figure 3.25).

1. *prodottiTradizionali*: it is a list of typical Trentino products organized alphabetically by name, category, and region of origin;
2. *ricette*: it is a collection of Trentino-specific recipes using regional foods (appetizers, first dishes, second dishes, desserts)
3. *ospitalita*: it contains a list of places related to catering in Emilia Romagna;

Initial considerations. The terms-documents matrix is built during the *DPP* phase. For this system, size is 22,598 terms for 3 documents. This system was introduced specifically to analyze how efficiently the integration process works. Indeed, the first two datasets (*prodottiTradizionali* and *ricette*) are correlated, whilst the third (*ospitalita*)is not related to either of them. We expect that only the first two can be merged theoretically.

Syntactic analysis. The dendrogram of the hierarchical clustering of the syntactic analysis shows what was expected, the correlation existing only between the documents *prodottiTradizionali* and *ricette* (see Figure 3.26-a). This result is also highlighted from the subsequent k-means algorithm (see Table 3.8), with a final fitting of 99.6%.

Table 3.8: K-Means results for Cooking System

Iter	Param k	Param N	Documents	Clusters	Bss/Tss	Doc Out
K-Means for Syntactic Analysis						
1	2	3	1,2,3	C_1={1,2}, C_2={3}	99,6%	
K-Means for Semantic Analysis						
1	2	3	1,2,3	C_1={1,2}, C_2={3}	78,7%	

1=ricette 2=prodottiTradizionali 3=ospitalita

Semantic analysis. Also in this case, the same cluster between *prodottiTradizionali* and *ricette* documents is clearly shown from the dendrogram of the hierarchical clustering, from the SOM training results (see Figure 3.26-b) and it is also confirmed by the subsequent k-means (see Table 3.8) that shows the same cluster with a final fitting of 78.7%.

Analyses comparison. The outcomes of the two analyses are almost equivalent. Therefore, if required to choose between them, the semantic analysis usually prevails because it is more accurate. In these conditions, a good choice of the parameters could be $\alpha = 0.35$ and $\beta = 0.65$.

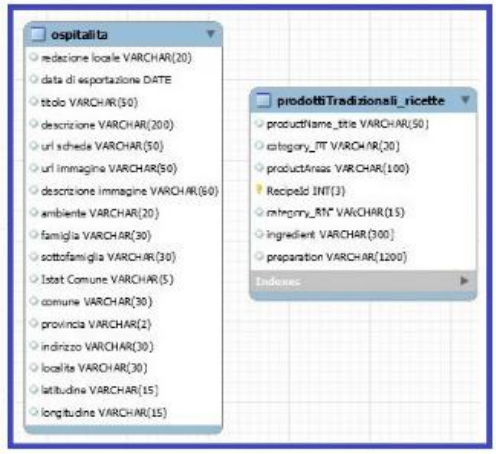


Figure 3.27: Reconciled schema for the Cooking System.

Final integration. The final merging of the tables *prodottiTradizionali* and *ricette* leads to an integrated table of 7 columns where only 1 is shared by the two tables (*title* for the first and *product_name* for the second). The sources tables had 3 and 5 columns, respectively. Figure 3.27 shows the reconciled schema.

3.5.4 *Final results comparison*

The outcomes for the four analyzed systems are presented in Table 3.9, where:

- **System** = name of the system under investigation;
- **Source Num** = number of system sources to be integrated
- **Source Tables** = number of tables with not only numeric data involved, for each of the starting sources;
- **Integrated Tables** = number of tables integrated for the system under investigation;
- **Fitting Syn/Fitting Sem** = best fitting indexes (BSS/TSS) achieved in the syntactic and semantic analyses;
- α, β = parameters decided by the Company Manager based on all available tools;
- **Integrated Tables size** = number of records of the integrated tables;
- **Accuracy** = measure of the quality of integration.

The accuracy measures the integration quality, evaluated by comparing the outcomes of the procedure outlined with oracles

Table 3.9: System test results table

System	Source num	Source tables	Integrated tables	Fitting syn Fitting sem	α β	Integrated tables sizes	Accuracy
CarShopping	2	3-5	2	82.5% 82.1%	0.40 0.60	A=122 B=122	100%
Panda	2	4-5	2	91.9% 95.2%	0.45 0.55	A=54458 B=54110	99%
Plants	2	6-3	2	88.7% 90.2%	0.45 0.55	A=32 B=14	17%
Cooking	3	1-1-1	2	99.6% 78.7%	0.35 0.65	A=109 B=49	100%

created by experts in the sector. According to our experiments, the stated procedure is valid for starting data that is consistent between 99% – 100%. We have taken into account the Plants system as a case with a high amount of inconsistent starting data, to be exhaustive. In that situation, the accuracy achieved is 17%. This outcome is the result of a syntactic anomaly in one of the columns that the process mapped and is not a process mistake. In this case, due to the too-small size of the source tables, the integration process resulted in a reconciled table that only contained one tuple.

3.6 CONCLUSIONS

The methodology described in this study concentrated on the DI of heterogeneous database sources, taking advantage of DM and ML techniques, and introducing the role of the company manager as an intermediary between the IT system designers and its final users. It only has a very minor role in this semi-automatic process, which is related to the creation of constraints and tolerance thresholds.

For systems made up of consistent data, the accuracy of the obtained results ranges from 99% to 100%, which is highly satisfactory. They are fairly comparable to what a developer would create if they followed the methods described in the literature. Furthermore, the consistency is preserved, making it possible to integrate the resulting schemes with the remaining tables from the sources that are not subject to integration. On the other hand, our method produced an accuracy of 17% for a system made up of inconsistent data sources.

4

A COMPARISON OF METHODS FOR THE EVALUATION OF AUTOMATIC TEXT SUMMARIZATION TECHNIQUES

In this chapter, we discuss about how the output of the ATS algorithms should be evaluated. The literature has proposed a variety of metrics and scores, but ROUGE is the one most regularly utilized to accomplish this goal. Our research study, through in-depth analysis of different datasets, revealed that this metric does not produce noteworthy results, due to the fact that ROUGE behaves similarly on EATS and AATS algorithms. This result is confirmed also restricting the starting dataset to a narrow interest field. Moreover, a cascade execution of two ATS algorithms produces better results with this metric. Finally, by comparing the findings of the ROUGE metric with those obtained from other frequently used ones, we concluded that there is still much work to be done before a suitable metric can be found to evaluate summaries produced by machines.

We start with an introduction to ATS algorithms, highlighting the still open challenges for their evaluation (see Section 4.1). Next in Section 4.2 the basic concepts related to text representation in data analysis and text similarity are introduced. The section 4.3 instead shows the State of the art related to the different methodologies of the ATS algorithms, then analyses the various approaches used in this research and finally gives an overview of the different evaluation metrics for the produced summaries. The experimentation is subsequently described and analyzed in all its details in two parts. The first is discussed in Section 4.4 and the second in Section 4.5. The threats to validity are explored in Section 4.6 and finally, the last section summarizes the findings and offers some suggestions for further perspectives (see Section 4.7).

4.1 INTRODUCTION

In recent years, the amount of digital content being converted into text has significantly increased. This trend (called *"infobesity"*) has driven researchers to try to extract as much knowledge as possible from it, through a variety of Data Mining (DM) techniques and data analysis approaches. ML and DL techniques, in particular, start to be the reference analytical methodologies to look for correlation in data, hidden patterns, and syntactic and semantic analysis as the amount of data increases. But it is a very difficult matter, due to the computing resources required and the complexity of the problems that need to be solved [9]. The primary issue is ensuring semantic coherence during information extraction, which is especially important considering the enormous volume of data available.

The development of ATS required over 50 years of research. Over time, a number of approaches for extracting a summary from a single document or a set of linked documents have been presented, and each of these techniques has produced highly efficient outcomes. ATS should offer users a simplified, brief, fluent, and quick understanding of information, presenting them in a compact form, by saving the key concepts from a source text and ensuring semantic coherence during information extraction at the same time. The objective of ATS methodologies is to accomplish this by creating algorithms aimed to generate summaries that include all pertinent information about a topic. The most important factors to take into account whilst creating a coherent summary are length, writing style, and grammar [1].

According to *Kucer* [59], humans who are summarizing a text must perform a cognitive process connected to comprehending the meaning of the text and they do not approach this activity uniquely. This is evident if the same individuals are monitored at different times [69] because the findings demonstrate that even among the same individuals, summarization outcomes might vary significantly in these cases. This concept is extremely important because, despite the fact that ATS has benefited from cutting-edge technologies like information retrieval and extraction [31],

1 http://www.hunter.cuny.edu/rwc/handouts/the-writing-process-1/invention/Guidelines-for-Writing-a-Summary

NLP [43] and ML [102], many questions about the relevance of a summary still remain unanswered. There are two main categories of algorithms for obtaining a summary from a text [32].

- *Extractive algorithms*: they select sentences from the input text, eliminating redundancy and choosing those that best cover all the important information;
- *Abstractive algorithms*: they attempt to create a new corpus, using different and more relevant terms and a different semantic composition in order to produce a simpler text.

Clearly, AATS methods are far more innovative than EATS ones. The creation of new ATS algorithms using both supervised and unsupervised algorithms [26] is a topic that is covered in the literature more frequently than the methods used to evaluate these algorithms, but it is really important to determine and confirm how accurately a summary reflects the original document. But in order to test the quality of a result, it is required to look into ATS algorithm evaluation techniques. There are several questions that require an answer, including:

- How can we determine which summary is preferable to another in an objective way?
- Exists a best-practice summary for any document created in accordance with a set of rules?
- What standards are used to evaluate this level of quality?

Determining the quality of the summaries produced by these methods is therefore the main challenge. Among the many metrics and scores discussed in the literature for this topic, ROUGE is the most popular used one [70]. It ignores the semantic and syntactic precision of the system and human summaries in favour of the overlapping of n-grams (represented as a numeric value). Therefore, the aim of this study was to correctly estimate the performance of the ROUGE metric. Since the EATS techniques, as previously indicated, use chunks of the original text to create a summary, whereas the AATS ones tend to introduce new words, the former should perform much better with ROUGE, as there may be more overlapping of n-grams [79].

In order to demonstrate this, we carried out an initial investigation through an experiment with the purpose of evaluating

the efficacy of this metric in judging the EATS and AATS algorithms, and then we ran a second experiment to compare the score obtained for two different ATS strategies: a simple execution of an ATS algorithm versus a multiple execution of different ATS algorithms on the same text. Our study revealed that ROUGE does not obtain first-rate results, as it gives a similar score both to EATS and AATS approaches; in addition, multiple executions are typically more advantageous than a single one.

In order to further our analysis, we looked at the outcomes of the ROUGE metric results on a variety of large datasets that were frequently used in the literature for the ATS tasks. Even after limiting the input dataset to specific fields of interest, we still attempted to determine whether the findings were solid. In order to compare the results of the ROUGE metric, some others metrics were taken into account. According to the conclusions, choosing an appropriate criterion to evaluate summaries produced by machines remains a challenging task.

4.2 BACKGROUND

4.2.1 *Text Representation*

A variety of the most popular methods for encoding information within a text are listed in [50]. Among the vectorial representations of a text, we find the *Bag of Words* and *Word embedding*.

4.2.1.1 *Bag of Words.*

Using this approach, it is easy to clearly identify the terms that are included in a document. This method is called "*bag*" because all data relating to the position of a word inside the text is no longer considered (see Figure 4.1). In practice, it concentrates on determining whether a word appears in the examined text or not. A vocabulary, or set of recognized words, expressed by a list of understandable phrases and a metric for their presence is required for this. The frequency with which the words appear in the vector is the used metric, where each location of the vector represents a unique word, and the number there indicates the

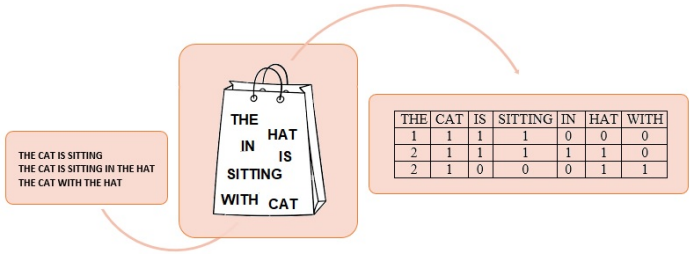


Figure 4.1: Bag of words representation.

total quantity of times that word appears in the text. The name of this kind of vectorial representation is"*sparse vector*".

4.2.1.2 *Word Embedding.*

Another method for encoding words in a multi-dimensional space is the Word Embedding (see Figure 4.2). It makes it possible to represent words with similar meanings in a similar manner so that they can be understood by ML algorithms. This approach is based on a highly computationally efficient data structure called "*dense vector*", where each word is represented by a vector of real values (usually of tens or hundreds of dimensions). Undoubtedly, acquiring this type of representation for each word requires a learning phase. Generally speaking, there are two categories into which word embedding falls, based on *Frequency* or *Prediction*.

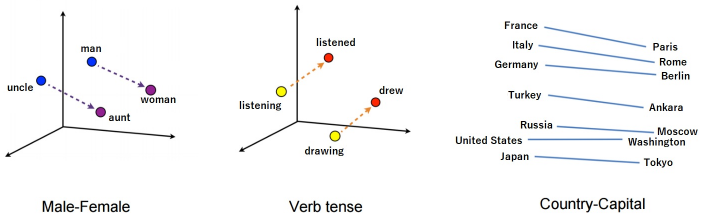


Figure 4.2: Multidimensional representation for word embedding.

Frequency Based Embeddings (FBE)
The FBE are distinguished mainly according to the type of vector that is built. We can have:

a) *Count Vector*: This matrix is built using a corpus of documents that contain tokens. Each row represents a document whilst each column represents a token since the dictionary will consist of the various tokens. So, both rows or columns can be considered vectors. This construction of the Count Vector can change depending on how the tokens that will be used are selected (for example all, or those that exceed a certain frequency and so on).

b) *TF-IDF Vector*: This approach, which also relies on the frequency method, differs from Count Vector in that it considers a presence of a word over the entire corpus rather than just in a single document. In a perfect scenario, we would like to give words that only exist in a subset of documents greater weight and less weight to common words that appear in nearly all texts. These common terms are penalized by TF-IDF by having a lower weight.

c) *Co-Occurrence Matrix*: The assumption behind this matrix is that words with similar meanings will frequently appear together and in contexts that are comparable, like in the case of the words "car" and "motorbike." The maximum distance between words on which we must base our assumption that two terms appear together in a given phrase must be established, in order to generate a co-occurrence matrix. The context window is the term for this distinction. Following the determination of this value, a matrix will be created, with each cell representing a numerical value based on the frequency with which the two matching terms have appeared in the same training environment.

Prediction Based Embeddings (PBE)
The PBE are distinguished in *CBOW* and *SKIP-GRAM model*.

a) *CBOW*: According to how CBOW acts, it tends to estimate the probability of a word within a context. Depending on

the context, a word or set of words may be used. The different vectors used to represent each sentence are identified, and these vectors are then given to a three-layer neural network. The weights of the network are modified using an appropriate procedure to produce the final output [146].

b) *SKIP-GRAM model*: The skip-gram architecture is the same as the CBOW topology but in reverse. This technique seeks to anticipate the context in which a word appears, as opposed to CBOW, which is concerned with predicting a word starting from the context. In this instance, a neural network is also used to drive all the actions [38].

Figure 4.3 represents the CBOW and SKIP-GRAM training approaches.

4.2.2 *Text Similarity*

Various *features* and *metrics* can be used to determine the degree of similarity between sentences. The majority of the time, linguistic and statistical computations are used to determine the *features* that can be extracted from the text. Term frequency, inverse document frequency, sentence position, sentence length, cue words, verbs, nouns, Part Of Speech (POS) and Named Entity Recognition (NER) tagging are some of the most frequently used in the literature.

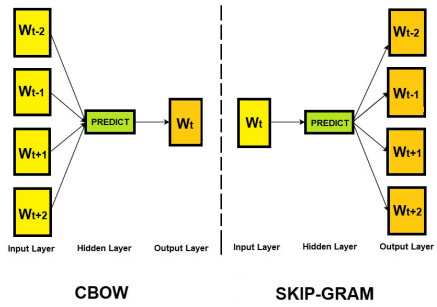


Figure 4.3: Approaches of training for CBOW and Skip-Gram.

Instead, it is essential to examine the possibility of quickly determining the coverage of relevant information, removing duplicate ones, when it comes to similarity *metrics*. Cosine Similarity (the most popular), Euclidean distance, Manhattan distance, and Jaccard distance are a few of the most popular distance metrics used in the ATS field [50].

- *Cosine Similarity* is a similarity measure between two non-zero vectors of an inner product space that measures the cosine of the angle between them;
- *Euclidean distance* is a measure of the straight-line distance between two points in Euclidean space;
- *Manhattan distance* is a distance metric that measures the absolute difference between two coordinates;
- *Jaccard distance* is a similarity measure between sets, defined as the size of the intersection divided by the size of the union of two sets.

4.3 RELATED WORKS

4.3.1 *Text Summarization methodologies*

When researching in the ATS sector, it is important to keep in mind Luhn's 1958 study [77]. It focused on creating summaries for scientific papers published in international journals. This research set the stage for all subsequent research and offered insightful advice on how to create algorithms for the automatic generation of summaries.

The assumption was that some words in a document reflect its specific text and that sentences in which these words are used in close proximity to one another are more informative of the meaning of the document.

In addition to extracting a summary in a broad context [6], ATS can be used to detect the topic of a text [63], discover and generate it [117], among other things. A third possibility is that the objective is to provide a summary based on a particular unique search engine queries of a user [3, 61].

Different summarization techniques have evolved over time on a variety of principles and are used in different scenarios. They take into account both the technical aspects of the generating technique and the decisions put forth by the application domain. According to the type of input, goal, and result, these categories are grouped (see Figure 4.4).

The type of input can be either singular or multiple: the former is used to create summaries from a single document, whilst the latter is used to summarize several documents, frequently on the same subject. The multiple documents ATS presents difficulties, particularly when data needs to be retrieved from the internet. In this case, documents related by one or more hypertext links[2] are combined into a single summary. A summary can be used also for a general or specific subject or to represent the findings of a search. Last but not least, the bulk of methods now in use can be categorized into two main groups: EATS and AATS methods [79, 90, 141].

2 unidirectional references introduced in an electronic document

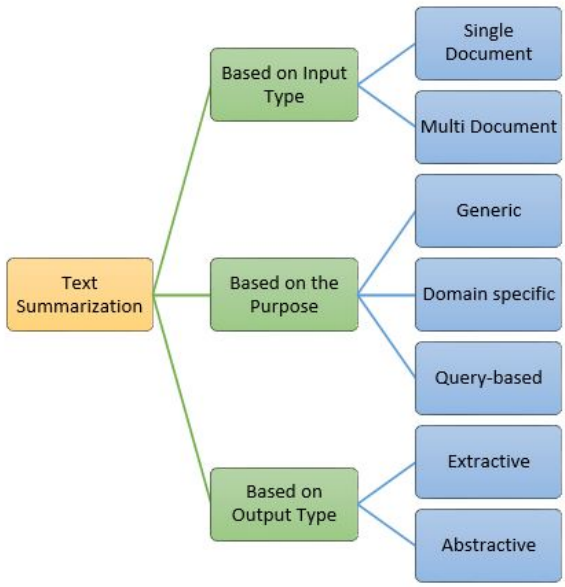


Figure 4.4: Text Summarization approaches.

Each phrase or paragraph from the source text is given a score in the EATS methods based on specific criteria, so to construct a summary by choosing each of them. The AATS approaches, on the other hand, make use of AI techniques. The input is reinterpreted to produce a summary that contains words and, in some cases, whole sentences which deviate from the original text [58].

Human judgement of the quality of a summary is subjective because it depends on individual relevance, comprehensibility, and readability standards, as demonstrated in [94]. In fact, a more in-depth research is carried out to strengthen our trust in our ability to automatically evaluate a summary produced by the machine. In many cases, the metrics employed for this purpose primarily provide a statistical approach to evaluation, comparing the overlap of words from both the summary and the reference text, but neglecting to analyze the semantic sense of what the text itself represents. Since the AATS methods use AI techniques that are more similar to what humans actually do when summarizing a text, they are obviously much more interesting than the EATS ones. Studies in the literature analyze in detail the algorithms and mechanisms on which they are based.

4.3.2 *Extractive Method strategies*

The attribution of a relevance index of sentences in a document, followed by the selection of the sentences with the highest scores and integrating them into a summary, is the basis of the EATS algorithms, according to Luhn [77]. Recent research has resulted in the development of distinct and increasingly complex EATS techniques, particularly in the AI field. In this section, several of the most promising methods in the literature, based on Clustering, Neural networks, Graphs, Fuzzy Logic and Query are explored.

The three key phases of an ATS task are the Data Pre-processing (DPP) phase, the *sentence scoring* phase, and the final *text extraction and summary producing* phase [55]. The DPP phase is fundamental since it is the starting point for the final outcome that a summary will produce. In this phase, stop-words are generally eliminated, and normalization, stemming algorithms, part-of-speech tagging, and tokenization are typically carried out. In the literature,

several methods for representing texts with the aim of ATS are offered, including the use of vectors and matrices to represent the features extracted from the text.

4.3.2.1 *Clustering Approaches.*

One of the most important aspects of ATS algorithms evaluation is to compare the number of topics the generated summary covers to the original text, in order to assess how good is the outcome. In fact, a text that needs to be summarized includes various different topics, and each of these is fully described in a range of different sentences throughout the document. Because of this, clustering-based ATS algorithms make use of clustering to identify the scores of the sentences, to find the topics of interest, or both in order to achieve the final result [20].

Alguliyev et al. [5] introduce *COSUM* (Clustering Optimization SUMmarization), one of the most recent EATS algorithms based on clustering and optimization methodologies. The selection of the topics of the input text and of the sentences to represent each of them in the final summary are the two key steps that make up the entire procedure. The k-means algorithm is used to address the initial phase. Then, each topic detected is associated with one or more sentences that represent it (each with a pre-established maximum length). Essentially, a sentence can only be a part of one single topic. The procedure used to choose the final summary sentences seeks to optimize the harmonic mean of the functions of the coverage and diversity of the sentences chosen for the summary. This approach aims to provide a summary of the input text that is compact, covers all the topics, and avoids using the same sentence more than once.

4.3.2.2 *Neural Network Approaches.*

These ATS approaches involve using a neural network to create a brief summary that captures the main points of the text. Networks can be trained on a variety of ATS tasks, such as keyword and sentence extraction for a summary generation. An overview of the most used algorithms is given in *Suleiman et al.* [130].

A *Restricted Boltzmann Machine (RBM)* [138] is a two-layered network of stochastic units composed of an input layer and

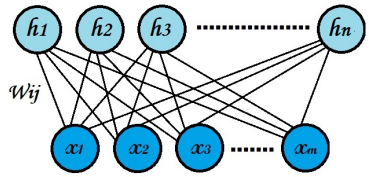


Figure 4.5: Restricted Boltzmann Machine (RBM) architecture.

a hidden layer, where connections are established exclusively between neurons in different layers (see Figure 4.5). In [142], a DL model is employed to produce a comprehensible and cohesive summary of factual reports. Instead of collecting features inside a corpus as other approaches do, the methodology extracts them individually from each input document. Then, to further enhance these features, each sentence is evaluated using a single RBM, which increases the accuracy of the resultant summary.

Likewise *Rezaei et al.* [112] develop a DL-based approach for multi-document ATS. In this work, two EATS approaches are described. The feature-sentence matrix was input into two different neural networks, a *9-layered neural network of autoencoders* (see Figure 4.6) and a *Deep Belief Network*, composed of several RBMs. After data normalization and feature extraction (the eight in-

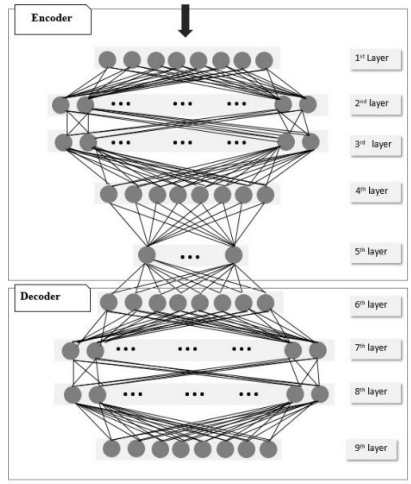


Figure 4.6: Autoencoder network architecture. (*Rezaei et al.* [112])

troduced features were determined for each sentence based on frequently used terms, phrase position, sentence length, and other factors), the relevance of the sentences was judged using the output scores of these networks. These DNNs have allowed for the improvement of the pre-existing matrix and the generation of more accurate summaries.

Instead, the *Variation Auto-Encoder* for an ATS challenge uses a neural network that consists of an encoder, a decoder, and a loss function. In this scenario, the encoder output serves as the decoder input and the decoder output is a probability distribution. During the DPP phase, the features of the text are extracted and a matrix made up of these data is created using statistical techniques or by counting the most commonly used terms. Each text is subjected to semantic analysis, resulting in the creation of a vector of features that will serve as input for the training phase. In the final stage, the sentences with the highest cosine similarity are selected.

An ATS methodology for summarizing a single document based on a RNN is presented by *Chen et al.* [22]. In this context, the RNN is made up of a number of hidden layers, each of which takes a list of words as input and outputs a summary of them. The encoder and extractor models serve as the foundation for the proposed network. The first chooses features using LSTM cells. The extractor employs a new reinforcement learning-based training approach to generate a weighted representation of each sentence in the input document, in order to select summary sentences more accurately and with more correlations (see Figure 4.7).

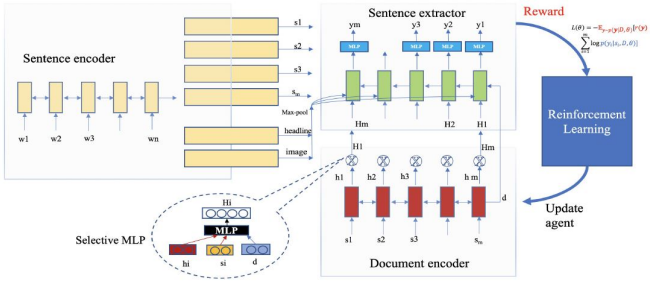


Figure 4.7: Model architecture. (*Chen et al.* [22])

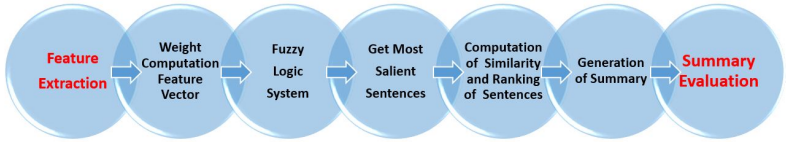


Figure 4.8: Summarization processing flow. (*Patel et al.* [101])

4.3.2.3 *Fuzzy Logic Based Approaches.*

Since *"fuzzy"* stands for hazy or unclear, *"fuzzy logic"* refers to all cases where it is impossible to know whether a statement is determined with certainty. So it serves as a method to normalize human abilities to reason abstractly. Therefore, instead of generating binary values, this logic takes into account real integers between 0 and 1 that represent "degrees of truth". ATS fuzzy logic approaches have four fundamental components: Defuzzifier, fuzzifier, fuzzy knowledge base, and inference engine [135]. The fuzzy system is fed information about textual properties, such as sentence length and similarity, using the fuzzy logic technique.

Patel et al. [101] propose a feature-based EATS algorithm, which uses a fuzzy model in order to obtain a final summary that has as much coverage as possible of the topics of interest of an input document. Using cosine similarity as a discriminant, the summarization phase reduces information redundancy by removing sentences that express the same idea. The final score of the candidate sentences is then calculated using the rule-based fuzzy inference method. Figure 4.8 illustrates the summarization processing flow.

In [123], a different EATS method based on fuzzy logic is provided. To improve the quality of the produced summaries of all lengths, the summary sentences are chosen based on a variety of factors, including the frequency of the terms and their position in the text.

4.3.2.4 *Graph Based Approaches.*

Graph-based methods approach the text from the viewpoint of a graph, where nodes stand in for sentences and arcs show

how similar or dissimilar they are, based on different criteria. The *PageRank* algorithm of Google [28], built on *Google's Hits* algorithms [97], have made significant advances in learning about the structure of the Web.

Two more widely used algorithms are *TextRank* and *LexRank* respectively. They are algorithms used to extract significant sentences from the text. *TextRank* [85] is a graph-based method that uses sentence similarity to rank key sentences, whilst *LexRank* [33] is a variation of the first that uses word similarities for the same aim. Both algorithms are unsupervised and can be used to identify key sentences in documents without any prior knowledge of the text.

In *Yasunaga et al.* [153], an innovative graph-based ATS system is presented. Sentence embeddings from RNNs are used as input node features, which uses a graph convolutional network (GCN) on relation graphs. The GCN produces high-level hidden sentence features for salience estimation through multiple layer-wise propagation (see Figure 4.9).With this methodology sentence relationships between documents can be captured.

A FrameNet-based Semantic Graph Model (FSGM) is suggested in *Han et al.* [41]. It considers sentences as vertices and semantic relationships as edges and gives weight to both. The algorithm discovers the semantic relationships between sentences, analyzing at the same time their meaning and their sequence of words, providing a better way to score sentences. The results of the experiment demonstrate that the model is very efficient for the TS task.

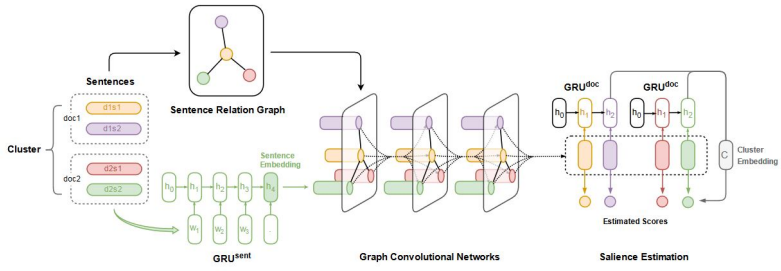


Figure 4.9: Architecture for sentence salience estimation. (*Yasunaga et al.* [153])

4.3.2.5 *Query Based Approaches.*

These types of text summarization methods are one of the most researched areas in NLP. They involve processing and understanding text documents with the appropriate output based on an input query [122]. These methodologies consist of selecting the relevant sentences linked to the input query and ensuring that they are in a readable form that the user can comprehend. All sentences in a document are ranked based on the frequency of their words. The score of each sentence is higher the more the phrase contains query sentences with respect to query terms. The output summary then includes the top-scoring sentences along with the structural context in which they were written.

In *Liu et al.* [74] an ATS framework based on a DL model is discussed. By choosing the appropriate set of sentences from a number of sources in line with the predefined query, this methodology aims to generate the final summary to ensure that it covers as many topics as feasible in relation to the original text. The methodology is broken up into three sections. The first one looks at finding the main topics in the input text, the second focuses on the generation of the summary and the third and final section talks about how to optimize the parameters for reconstruction using backpropagation on the entire deep model. The proposed framework successfully advances important concepts layer by layer, whilst retaining the unique mining DL capacity.

4.3.3 *Abstractive Method strategies*

The development of AI technology has led to the emergence of new summarizing strategies known as *Abstractive techniques*. In these techniques, the generated text is a revised version of the original one, built with new words and concepts [71]. This closely matches the way text summaries are written by humans. Basically, when a person reads a document, he first creates a semantic mental model of it. Then, using the words of his personal vocabulary, he selects those that satisfy the semantics requirements to create a concise summary that needs to include all the key concepts of the document.

Many approaches have been presented, with the *seq2seq model* and the *attention mechanism* serving as the foundation for the majority of them. Researchers have recently created a solid foundation for the creation of AATS algorithms, independent of the conventional approaches to NLP, following the increased implementation of neural networks and DL methods, which sometimes outperform extractive methodologies.

4.3.3.1 *Natural Language Generation (NLG) models.*

The larger goal of NLP is to create language models that are helpful for many ML and DL challenges, among which AATS stands out. Many improvements in NLP tasks have resulted from recent developments in pre-training language models on big text corpora because they have several benefits over structured knowledge bases [103]. They actually are simple to expand to more information and do not need to be trained by humans. Such a model is basically a probability distribution over word sequences. To do this, models must be trained on a set of samples made up of word sequences of various lengths. The hypothesis is that each word in a sentence is conditioned by the words that come before and after it in the sentence [105]. So, during training, this is taken into account. As a result, generally, two distinct models are trained before being combined to create a single probability distribution. But recent advancements have made explicit bidirectional conditioning possible [27] by masking the conditioned words. DL techniques have recently achieved very high performance in language modeling and are now a common tool for this purpose.

4.3.3.2 *Sequence to Sequence Model.*

The model named *Sequence to sequence (Seq2seq)*, is a special type of RNN architecture frequently used to address difficult language challenges. ATS is one of them. The input and output sequences for this *encoder-decoder* neural network model have different lengths. For the ATS task, the encoder looks at the full input sequence to produce a vector of features. The implementation of the encoder and decoder with the help of various types of neural networks has been implemented in a variety of ways in the liter-

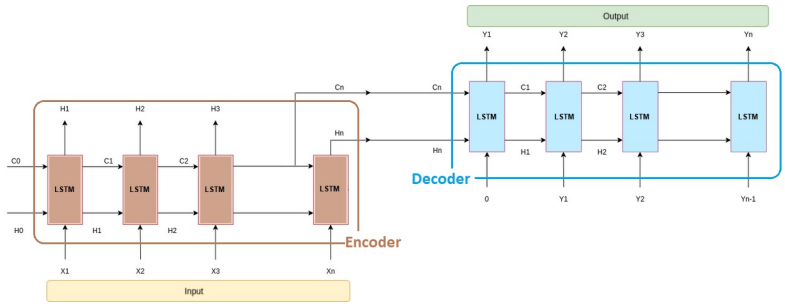


Figure 4.10: Encoder/Decoder representative models

ature. In essence, RNN [88], Gated Recurrent Unit (GRU) [66, 110] or LSTM [42, 128] are often used. The latter neural network is the most preferred since it enables the identification of long-term dependencies whilst avoiding the vanishing gradient problem [131]. Encoder-Decoder work can be divided into two distinct phases: training and inference.

a) *Training phase*: during this phase, a separate time frame will be used to train the encoder and decoder to predict the target sequence (see Figure 4.10);
   - *Encoder*: this module is composed of LSTM cells. Using an LSTM, the entire input sequence is read and a word is inserted into the encoder at any instant of time;
   - *Decoder*: this module also contains LSTM cells. It reads the whole sequence word by word, and attempts to predict the same sequence with the difference of an instant of time.

b) *Inference phase*: this phase involved evaluating the model whilst accounting for some new sequences for which the target sequence is determined. The steps it follows are:
   1. the decoder is initialized with the internal state of the encoder after the encoder elaborates the whole input sequence;
   2. the token *start* is transmitted as the first input of the decoder;
   3. the decoder is started for an instant of time at the time;

4. the output is the probability of the subsequent word;
5. the word with the highest probability is chosen and entered as input into the following instant of time;
6. during this period, the internal state of the decoder is updated with the weights of the new cells.

Steps 3–5 are executed until the token *end* is read.

4.3.3.3 *Attention mechanism.*

Considering the behaviour of these neural networks, it would be very important to understand how much attention each word of the input sequence should have in order to generate a good summary. This doubt underlies the *attention mechanism* that aims to mimic cognitive attention [93]. The fundamental idea behind this concept is to let the decoder use the most significant elements of the input sequence. In order to accomplish this, the input vectors are combined in a weighted manner, giving the more important vectors a higher weight than the less important ones. In literature, there are two classes of attention mechanisms that differ by the way the context is derived:

- *Global attention*: the attention layer is connected with all the hidden layers of the encoder, so are all considered in the computation of the context (see Figure 4.11-a);
- *Local attention*: the attention layer is connected only partially with the states of the hidden layer of the encoder. Only these

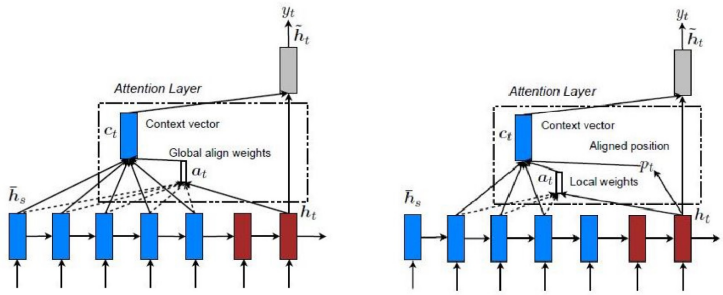


Figure 4.11: Attention Mechanism classes.

are used for the calculation of the context (see Figure 4.11-b);

To prevent the tendency of neural Seq2seq models to replicate and inaccurately reproduce factual components, *See et al.* [120] suggest a new approach that enhances the conventional Seq2seq attentional model. The skills of the *pointer networks* [143] serve as the foundation for a hybrid model that can properly reproduce information. This new model enables both word copying using pointing and word generation from a given vocabulary (see Figure 4.12). Additionally, it has the capacity to generate new words using a generator. Last but not least, employing coverage facilitates keeping track of what has previously been reduced, avoiding duplication.

4.3.3.4 *Transformer Network.*

Vaswani et al [140] have developed the *Transformer* encoder-decoder architecture, which is entirely based on the attention mechanism and completely eliminates recurrences and convolutions. Whilst in an RNN architecture, the input of a sentence is provided one word at a time to produce word embeddings, in a *transformer* all the words of a sentence can be passed at once, allowing for simultaneous word embedding determination. So, this work introduces a new methodology for modeling dependencies without taking into account where they occur in the input or output sequence. The *transformer* follows to the overall architecture of the

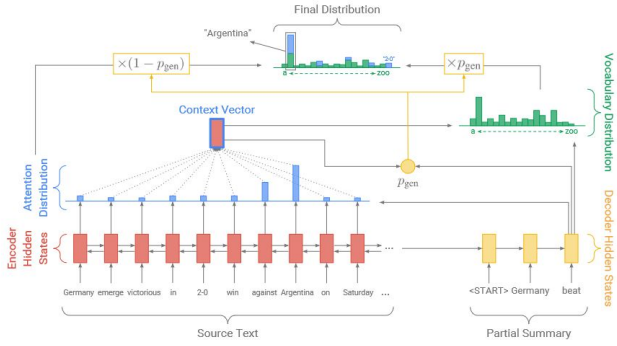


Figure 4.12: Pointer-generator model. (*See et al.* [120])

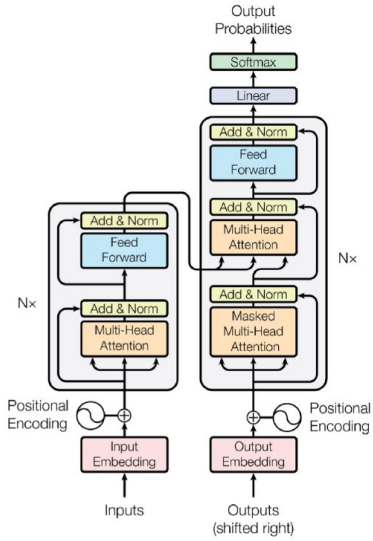


Figure 4.13: Transformer architecture. (*Vaswani et al* [140])

most efficient neural networks, made up of an encoder and a decoder, both composed of stacked layers. In this case, each layer is made up of a multi-head self-attention mechanism sublayer, and a position-wise fully connected feed-forward network sublayer. A third additional sub-layer of the decoder provides multi-head attention over the output of the encoder stack.

- **Encoder block**: here the input words are transformed into vectors (*Input Embedding*) and then processed to identify the relative importance of each word in the sentence (*Multi-Head Attention Part*). In order to quantify this relevance, an attention vector is constructed for each word. Due to their independence, these vectors can be processed in parallel (*Feed Forward Network part*), allowing all words to be sent to the encoder block at once and obtaining the set of vectors that have been encoded for each word at the same time (see the left side of Figure 4.13).

- **Decoder block**: compared to the encoder, the decoder has an additional level (*Masked Multi-Head Attention Part*) that hides the correct outcome starting with the output words (*Output Embedding*), allowing learning to occur without

conditioning by creating new attention vectors. These new vectors are subsequently passed into a second *multi-head attention block* along with those from the encoder output. After their processing, they finally pass to the *feed-forward* unit that transforms them for subsequent processing. The final *softmax* layer will convert the input into a probability distribution that can be decoded to produce the outcome of the prediction (see the right side of Figure 4.13).

4.3.3.5 *BERT.*

Due to the unidirectional nature of most conventional language representation models [140], there are not many pre-training approaches that can be used, especially for fine-tuning methods. To this aim is presented the framework BERT [27] as an attempt to pre-train a deep bidirectional Transformer. Its basic idea is to predict the original vocabulary id of a hidden word only from its context, by using a masked language model that randomly disguises some of the input tokens. It works in two distinct phases, *pre-training* and *fine-tuning* (see Figure 4.14).

- *pre-training phase*: two unsupervised approaches are used instead of the algorithm being pre-trained with one-way language models. The first approach (*Masked Language Model*) involves randomly masking 15% of each wordpiece token in each sequence before predicting them. This methodology is used to train a deep bidirectional representation. The process also involves randomly substituting the "masked" words with the [mask] token, a random token, or the original token; after that, the model is trained to predict the

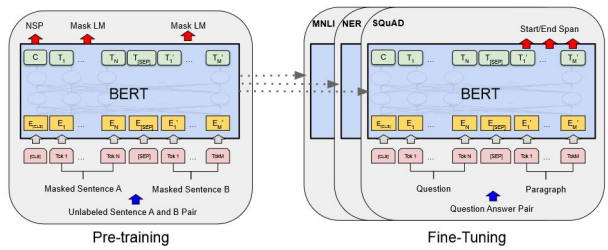


Figure 4.14: Bert pre-training and fine-tuning. (*Devlin et al* [27])

original token using cross entropy loss. The second approach (*Next Sentence Prediction*), trains a model that can recognize sentence relationships. With the help of this challenge, a model is pre-trained that will be used for downstream tasks like question answering and natural language inference. Although it shares many similarities with other representation-learning goals, it is different in that it transfers all parameters to the end-task model for subsequent tasks.

- *fine-tuning phase*: with the BERT architecture, fine-tuning a pre-trained model is a quick and simple operation, because it can perform a variety of tasks, including paraphrase, entailment, question-answering, and text classification, thanks to the self-attention mechanism of the transformer. All model parameters are fine-tuned end-to-end after the task-specific inputs and outputs are plugged in. This process is relatively inexpensive and can be replicated with a few hours on a GPU or less than an hour on a cloud TPU.

4.3.3.6 *Approaches used in the literature.*

The Seq2seq model forms the foundation of many of the approaches investigated in the literature, but each one adds some modifications and enhancements to improve its efficacy.

Yang et al [152] suggests using the model with two encoders: the first computes the semantic vector for each word in the input; the second calculates the weight for a word and then recalculates the corresponding semantic vector, producing a more fine-grained encoding based on both the input raw text and the previously generated text summarization output. Finally, a decoder is tasked with producing the final output.

For the purpose of automatically summarizing extensive texts, *Li et al* [65] suggest an architecture of Seq2seq with the inclusion of a multi-head self-attention mechanism (MHAS). In order to avoid producing a list of words that are redundant repetitions of previous ones, this form of attention mechanism computes n functions of attention in parallel before merging them. This approach improves how well the model maintains the original data, producing positive outcomes.

In comparison to previous methods, *Rekabdar et al* [111] proposes using Generative Adversarial Networks (GAN) [49] to create a summary that will be difficult to differentiate from one created by a human. In particular, the author suggests a network (*generator*) that generates the summary and a second network (*discriminator*) that seeks to figure out the probability that a summary originated from training data rather than the generator in order to direct the training of the model. It can be noted that the networks that make up the generator employ a model that is based on the encoder-decoder with LSTM cells design. In order to increase precision, they also employ an attention mechanism.

4.3.4 *Text Summarization algorithms*

Below is a brief overview of the ATS algorithms used in this research study.

The most significant phrases (or portions of phrases) in the text to be summarized are found and then exactly reproduced in EATS techniques. Since what is already in the document is reused, no new text is produced [87]. All *Extractive* methods vary considerably, but they involve the same fundamental tasks of building an intermediate representation of the input text to be summarized, scoring the sentences based on the intermediate representation and choosing the first n sentences with the highest score, using some evaluation methods, as final summary [55]. This research study specifically took into account four EATS algorithms: *Textrank*, *Lsa*, *Luhn* and *Lexrank*.

4.3.4.1 *Textrank.*

TextRank is one of the extractive unsupervised algorithms we have considered. It starts from some articles that represent a specific argument and tries to produce the best summary of the argument [23]. Specifically, once all the pages of the document to be summarized have been concatenated, all its single phrases are recovered to extract the vector representation of each of them. These vectors will be used to generate a similarity matrix, which is then transformed into a graph, where the nodes and edges correspond to the phrases and the similarity scores respectively,

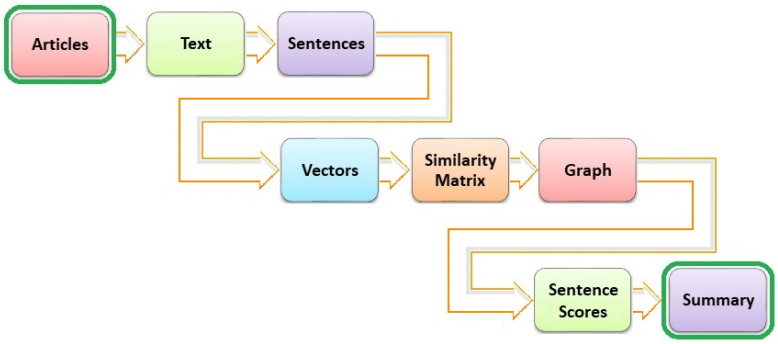


Figure 4.15: TextRank algorithm.

to establish the precise weight of each sentence. Finally, the most pertinent phrases that will make up the final summary of the text will be picked based on these values (see Figure 4.15).

4.3.4.2 *Lsa.*

The *Lsa* [95] algorithm is based on the presumption that similarities between terms and concepts can be identified since words frequently have similar meanings when they are employed in similar settings. This algorithm starts from a document-term matrix where each cell will contain the frequency of the term (*column*) for the specific sentence (*row*). Using the Singular Value Decomposition (SVD) [2], one of the dimensionality reduction methodologies, that provides a base for the LSI used to analyze the relationship between a set of documents and terms contained in each of them, this matrix is decomposed (see Figure 4.16). The decomposition allows having in a matrix the weights for each topic-related word, in a second one the weight of each topic and in a third the weight of the documents of each topic. The core element of SVD is that the document-term matrix may be represented as vectors points in a Euclidean space. So the documents or sentences in our scenario are displayed in this space using these vectors. The weighting of the sentences corresponding to various topics is determined by the LSI. Indeed, depending on the weight of each topic, the top-ranked sentences are chosen for summarization.

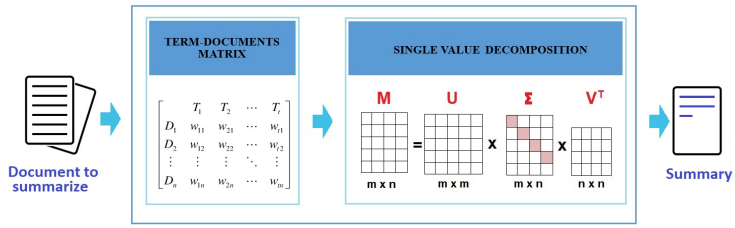


Figure 4.16: Schema of Lsa algorithm.

4.3.4.3 *Luhn.*

The premise behind the heuristic method of *Luhn* for summarizing text [76] is that the importance of each word in a document is established by a significance rate. In particular, *Luhn* considers unimportant the words with a too-high frequency rate and those with a too-low frequency. Thus, making a descending list based on the frequency of each word, Figure 4.17 shows that the words that can be considered as *"significant"* will be those between the two limits A and B. The above-mentioned frequency rates are represented by these restrictions (too high and too low). In practice, the more frequently particular words appear together in a sentence, the more meaning each of these words can be assigned.

The *"significance factor"* that arises from what has been mentioned, represents the number of occurrences of significant words within a phrase and the linear distance between them, as a result of inconsequential terms like *stopwords*. Based on this principle, all sentences in a text can be sorted by their meaning, and one or

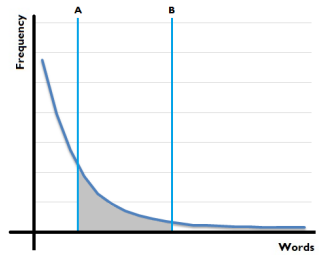


Figure 4.17: Words-Frequencies Diagram.

more of the best-scoring phrases can then be chosen to serve as a summary. The reason for which the frequency of terms in a text is used to give *"importance"* to them, is related to the fact that people who express a concept usually speak in circles around the same term, occasionally employing synonyms that are limited. This is essentially a measure of the relevance of a term.

4.3.4.4 *LexRank.*

LexRank is an unsupervised method for summarizing text that uses graph-based centrality score [33]. The essential idea is that phrases "recommend" additional sentences that are similar to them. In order to rank highly and be included in a summary, a statement must be comparable to many other sentences. First, the sentences must be transformed into real number vectors. The similarity between the different vectors is estimated and organized in a *adjacency matrix*, where each row and each column represent a single phrase and each cell provides their similarity value. This is done using the *cosine similarity* as the metric. At this point is used a *connectivity matrix*, which differs from an adjacency matrix, because it is a list of which vertex have an edge between them in the graph that represents all the edges between sentences. Finally, *LexRank* makes use of eigenvector centrality to start from this matrix and determine the most significant sentences by the Power Iteration Method [17].

AATS algorithms must comprehend the semantics of documents before employing the NLG process to provide a more succinct summary. Traditional approaches to ATS typically rely on manual methods to extract features from the text [67], which is time-consuming, particularly for very large textual corpora. These features have become automatically generated, thanks to DL during a training phase of a DNN on the available data.

The conceptual model of an AATS method is briefly depicted in Figure 4.18. The input documents that need to be summarized are preprocessed by various operations, like cleaning, data splitting, tokenization, etc., to make them ready for processing. The use of *Word Embedding* methods [96] in correlation with a DNN, the semantic understanding phase is performed to identify and

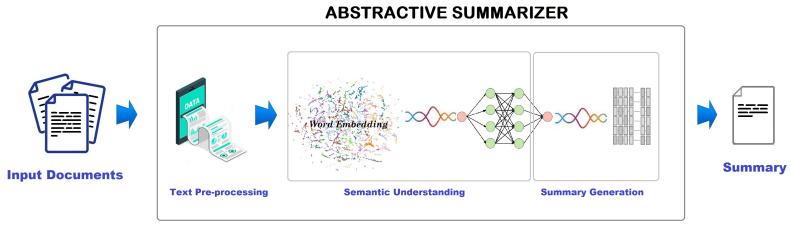


Figure 4.18: Abstractive Text Summarization algorithms.

represent the semantics of the text document. This process takes place in the vector domain and produces a fusion vector that is modified for the generation of the final summary, adopting synonymous substitution, paraphrasing, sentence reductions and so on.

A *Seq2seq* model with attention has been adopted. The *encoder-decoder* network is trained using labels and targets consisting respectively of the original texts and their summaries of a specific dataset. The vectorization of such labels and targets will form a vocabulary. At this point, by using *word embeddings* for the words in the vocabulary, it is possible to adapt the labels and objectives to the training model. The *attention mechanism* has the aim to refine the work, focusing only on a few chosen input sequences at a time, rather than the whole sequence, to predict a word. This research study specifically took into account *Word2vec*, *Doc2vec* and *Glove* embeddings. The three models are trained in distinct ways, so they result in slightly different properties in the word vectors that they generate.

In fact, it must be emphasized that the outcomes obtained by the various approaches, are more affected by the datasets used to train the various models, the length of the vectors, and the application settings, rather than the type of model.

4.3.4.5 *Word2vec.*

Word2vec[86] is mainly based on the training of a shallow *feed-forward neural network* [118]. In particular, this technique analyses a collection of input texts and *"vectorizes"* its words, using a two-layer neural network, that can be trained both through the

Figure 4.19: Training architectures of CBOW and Skip-Gram.

CBOW or the Skip-Gram model (see Figure 4.19). Based on the co-occurrence of words in the dataset, vector representation derives semantic associations. The produced output is a vocabulary where each word is associated with a number vector. Obviously, the model will be much more accurate in predicting words, the more times these are identified in the same context.

4.3.4.6 *Doc2vec.*

Doc2vec [64] is similar to *Word2vec*, but it uses a document feature vector in addition to word vectors to predict the next word. In simple terms, the document vector Par_{id} is trained together with the word vectors w_1 through w_n of the document, and at the conclusion of training, it contains a numerical representation of the document. It is important to note that an embedding can be created using documents of any length. Two topologies can be used to construct these ***P**aragraph **V**ector* models: the ***D**istributed **B**ag of **W**ords* version (DBOW-PV) and the ***D**istributed **M**emory* version (DM-PV). Figure 4.20 shows the *DM-PV* that operates as a memory that recalls what is absent from the context. In

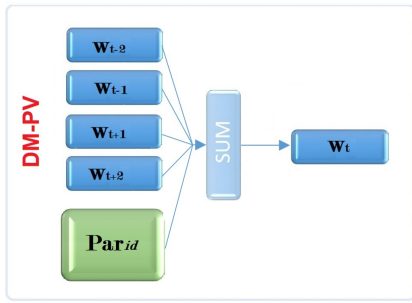


Figure 4.20: Training architectures of DM-PV.

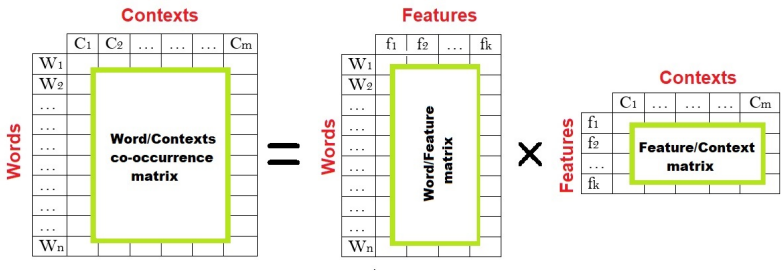


Figure 4.21: Embedding Matrix in Glove.

summary, a set of documents are needed in order to train this model that produces a word vector w_i for every word and a document vector Par_{id} for every document.

4.3.4.7 *Glove.*

The *Glove* model is based on *matrix factorization methodologies* [126]. The foundation of this model is the use of global word-to-word co-occurrence counts throughout the entire corpus. In the beginning, a huge matrix of co-occurrence data is built. Each cell of the matrix represents the frequency of the word (row) in a specific context (column). Obviously, the number of contexts is high, given that it essentially has combinatorial dimensions. This matrix is then factored in order to make it smaller, so as to represent by a vector, the various features of each word (see Figure 4.21), where the distance between words will be related to their semantic similarity.

4.3.5 *Summary Evaluation Methods*

The evaluation of a summary by a human is a very common task. In reality, by reading and comparing two documents, one can assess the quality of the summary by analyzing which of them is more specific or covers the more important concepts, or if one is more readable and grammatically correct than the other.

An AATS method, as compared to an EATS one, creates a summary that is more similar to what a human would write. This is because a person can attempt to explain his ideas using new

Figure 4.22: Evaluation methods overview.

terms or sentences after reading and fully comprehending one or more source documents, and making sure not to exclude any important points from the summary. This kind of job demands a lot of creativity, and individual results may differ substantially. Since a concept can be expressed in a variety of ways using different phrases and words, lexical construction is essential.

Figure 4.22 illustrates the various evaluation criteria, which are divided into intrinsic and extrinsic categories. This section focuses on intrinsic ATS evaluation methods, including a detailed analysis of the most used metric, the ROUGE.

4.3.5.1 *ROUGE.*

ROUGE is the acronym for "*Recall-Oriented Understudy of Gisting Evaluation*". It refers to a set of criteria for evaluating texts that are produced automatically. It is typically employed to determine the effectiveness of an ATS summary.

ROUGE works by comparing a machine-generated summary (also known as a *system summary*) with one written by a human (sometimes called *gold standard* or *reference summary*). In particular, it counts the number of *n-grams* that matches the machine-generated text and a reference one (where an *n-gram* is just a collection of tokens or words). Based on n, it is possible to have *unigrams* (which are made up of a single word), *bigrams* (which are made up of two words), *trigrams* (which are made up of three words) and so on (see Figure 4.23).

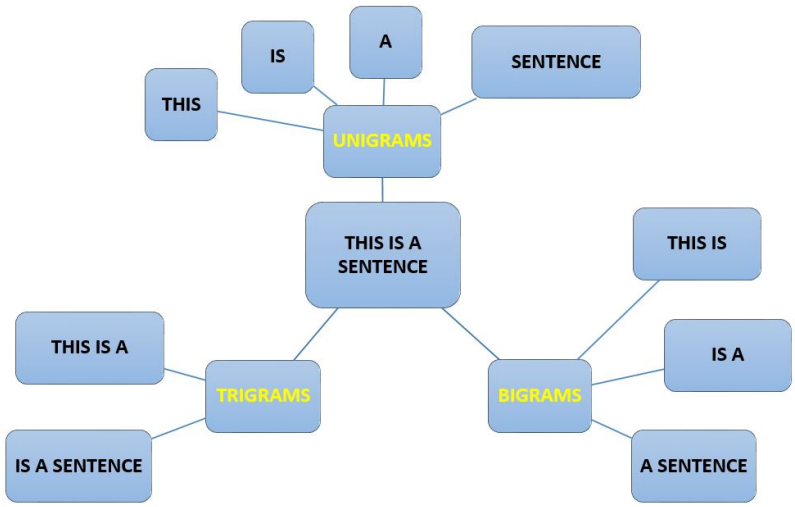


Figure 4.23: N-grams in a sentence.

We can use *recall*, *precision* or *F1-score* measures to evaluate this metric (also if the *recall* is the most considered). *Recall* is defined in Equation 4.1 as:

$$ROUGE\ recall = \frac{num\ of\ overlapping\ n\text{-}grams}{num\ of\ reference\ summary\ n\text{-}grams} \tag{4.1}$$

The *recall* aims to quantify how many n-grams in the referenced summary were captured from the summary output. It is important to underline that due to redundancy, a machine-generated summary can be very long. In fact, whilst a human can easily remove redundant parts from a document, automatic tools could have to deal with an excessive amount of terms of reference, making the summary excessively long. As a result, the *precision* tries to capture how short the summary of the system is, and how it avoids using unnecessary words in its corpus. It is defined as follows (Equation 4.2):

$$ROUGE\ precision = \frac{num\ of\ overlapping\ n\text{-}grams}{num\ of\ system\ summary\ n\text{-}grams} \tag{4.2}$$

Last but not least, the *F1-score* can be used to combine *precision* and *recall* to quantify the accuracy of this measure. It is calculated as follows (Equation 4.3):

$$ROUGE\ F1 = 2 * \frac{precision * recall}{precision + recall} \tag{4.3}$$

But when we discuss the ROUGE metric, we do not just mean one particular sub-metric; rather, we mean a collection of several. According to the different granularities, ROUGE metrics can be computed using a variety of approaches. The following are used the most frequently:

1. ROUGE-N refers to the overlapping of N-grams (unigrams, bigrams, trigrams and so on) between the system summary and the reference summary;
2. ROUGE-L measures the longest common word sequence, computed by the Longest Common Subsequence (LCS) algorithm;
3. ROUGE-S refers to a couple of words in an ordered sentence, that allows some gaps. Sometimes this measure is also called skip-gram;
4. ROUGE-SU is a weighted mean between ROUGE-S and ROUGE-L.

The most widely used ROUGE typologies in the literature are ROUGE-1, ROUGE-2, and ROUGE-L because they accurately reflect the granularity of the texts under investigation.

4.3.5.2 *BLEU.*

The BLEU (*BiLingual Evaluation Understudy*) [98] metric was created to evaluate the quality of a translation compared to a reference translation. But due to its properties, it is also widely used for the evaluation of ATS algorithms. Its purpose is to give each summary a specific numerical score that indicates how *"good"* it is in comparison to a reference summary. The basic strategy used by this metric is to count the overlapping of *n-grams* between the

machine-generated summary and the reference summary. If we consider the *precision*, we have (Equation 4.4):

$$BLEU\ precision = \frac{num\ of\ overlapping\ n\text{-}grams}{num\ of\ system\ summary\ n\text{-}grams} \tag{4.4}$$

In general, the *precision* ranges from 0 to 1, and higher scores indicate a better summary. This metric uses a few strategies to overcome some current problems of n-grams-based evaluation metrics:

- The fact that summary generation models occasionally repeat the same n-gram more than once can lead to high *precision* values if is simply counted the number of n-grams matches. To address this issue, BLEU employs a modified *modified precision* that takes into account the number of times an n-gram is repeated in the system summary in relation to the maximum number of times it appears in the reference summary.

- The fact that *precision* ignores the order of n-grams in summaries is another problem with it. In fact, its value on a summary with n-grams in the opposite order of those in the reference one would therefore be high. To fix this issue, BLEU estimates the *precision* for a number of n-grams before averaging the results. The geometric mean of all the scores is then used to compute the final value. Usually are taken into account n-grams with n varying from 1 to 4.

- Additionally, there is a *penalty* for brevity. When the system summary length is identical to the reference one length, the *Brevity Penalty* (BP) will be set to 1. The following Equation 4.5 is used to determine BP, which decays exponentially:

$$Brevity\ Penalty = \begin{cases} 1, & if\ cSS > cRS \\ e^{1-\frac{cRS}{cSS}} & if\ cSS \leq cRS \end{cases} \tag{4.5}$$

  where cRS and cSS are respectively the counts of words in the reference summary and in the system summary

The BLEU metric is not a perfect metric, despite having a number of interesting properties. It really has a number of problems, including the fact that it ignores the semantics of sentences and is unaware of synonyms and paraphrases.

4.3.5.3 *METEOR.*

METEOR (*Metric for Evaluation of Translation with Explicit Ordering*) is a metric for machine translation evaluations [11], but for its intrinsic features, it is also used for the evaluation of ATS algorithms. It tries to fix what is missing in the BLUE metric. In fact, the latter heavily relies on *precision* and completely ignores the *recall*, or in other words, whether the system summary accurately captures all of the information in the reference summary. It also disregards the semantic similarity of the sentences. In this context, it can be thought of as a quantitative method for evaluating system summaries, based on a generalized idea of *unigrams correspondence* between machine-generated texts and human-generated references. This metric comes closer to what a human would judge to be a good evaluation criterion since it weighs *recall* more heavily than *precision* and also takes into account semantic similarity by employing a stemmer or synonym matcher to find words that are related to one another. Additionally, when there is a grammatically correct match, it provides more weight. The primary and derived forms and meanings of unigrams can be mixed. METEOR uses a combination of *unigram-precision* (see Equation 4.6), *unigram-recall* (see Equation 4.7), and a *fragmentation measure* (see Equation 4.8)to determine a score for this mixture.

$$P = METEOR\ precision = \frac{num\ of\ overlapping\ unigrams}{num\ of\ system\ summary\ unigrams} \tag{4.6}$$

$$R = METEOR\ recall = \frac{num\ of\ overlapping\ unigrams}{num\ of\ reference\ summary\ unigrams} \tag{4.7}$$

$$F_{mean} = \frac{10 * P * R}{R + 9P} \tag{4.8}$$

The F_{mean} score is the harmonic mean of *precision* and *recall*, and it gives *recall* more weight than *precision* because (R + 9P) is included in the denominator.

The current methods only take into account concordance between single unigrams, not between longer parts that are present in both the reference and candidate sentences. Extended n-gram matches are utilized to compute a *Block penalty* for the alignment that takes these into consideration. This penalty (see Equation 4.9) will be increased in proportion to the number of mappings between the reference and candidate sentences that are not contiguous.

$$penalty = 0.5 * \left(\frac{c}{u_m}\right)^3 \tag{4.9}$$

where c and u_m are respectively the number of groups in the system summary and the number of unigrams in the reference summary. The Equation 4.10 combines F_{mean} with this penalty, and gives the final METEOR score that is:

$$M = F_{mean}(1 - penalty) \tag{4.10}$$

4.3.5.4 *Pyramid.*

Pyramid is suggested as an innovative method for evaluating texts that were created automatically [91]. The basic concept is based on the identification of some little units, each called Summary Content Unit (SCU) that will be used to compare the information in the original text.

Each SCU is a brief piece of text that is no longer than a sentence. Depending on how frequently it appears in the different texts under consideration, it is given a weight. A small number of SCUs with a heavy weight and an increasing number of SCUs with a light weight are both reasonable expectations. This approach performs well when there are several texts from which SCUs can be produced so this structure suggests the name of the method because it is hierarchical.

The building of the pyramid is shown in Figure 4.24. The approach can be synthesized in the following steps:

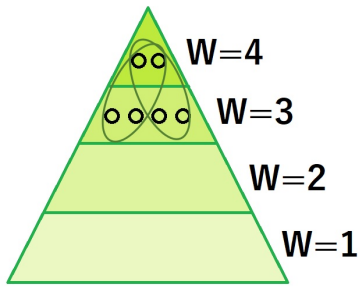


Figure 4.24: The Pyramidal hierarchy.

1. **Enumeration**: in this phase are listed all the SCUs of the sentences from the peer summary;
2. **Pyramid generation**: each SCU is partitioned using a pyramidal hierarchy scheme, where every level contains acSCUs having the same weight. A pyramid was also created for the reference summary;
3. **Scoring**: The ratio between the total weights of the SCUs in the system summary and the reference summary is determined. The resulting values vary from 0 to 1, where 1 represents a high weight for the majority of the information.

Even though this method seems to be very useful in evaluating the summary produced by ATS algorithms, it has not been as effective as the ROUGE metric.

4.3.5.5 *Semantic Similarity for Abstractive Summarization (SSAS).*

Unfortunately, statistical techniques are unable to identify semantic contradictions within a text or other aspects of natural language, such as paraphrasing and logical consequences. Therefore, various approaches to address this problem have been put out in the literature, particularly for the AATS algorithms. Actually, SSAS is a metric that highlights the semantic connections between the system and the reference summaries.

Here we can see a general overview of how the SSAS technique is used to create a score:

- First, the text is cleaned up by removing all SCUs from both the system and reference summaries. The cardinality of the two sets will be n and m, respectively;
- Following that, a range of natural language features, including inference, are extracted from the corpus. To determine a final score, a classification inference model is utilized to train the weights of different combinations of these characteristics.
- Finally, the findings are normalized, and they are ranked.

However, due to its computational complexity, this strategy, like Pyramid, also failed to acquire traction in the literature.

4.4 FIRST INVESTIGATION

Seven ATS algorithms were investigated, involving four Extractive (*textrank*, *lsa*, *luhn* and *lexrank*) and three Abstractive (*word2vec*, *doc2vec* and *glove*).

4.4.1 *Dataset*

All experiments were performed on the **CNN/Daily Mail** dataset[3], the most commonly used by researchers to test novel summarizing techniques, which consists of CNN and Daily Mail articles and editorial news. This dataset, which was first introduced for AATS, has roughly 287.000 articles, including summaries, and is one of the most frequently used datasets for ATS algorithm evaluation [88];

4.4.2 *Research Questions*

The first investigation aims to respond to two specific Research Question (RQ)s. The first is inspired by an extensive literature analysis that has shown some ambiguities regarding the most often used criterion to evaluate the accuracy of ATS algorithms [92]. In this regard, the investigation related to the first RQ focuses

3 CNN Daily-Mail: https://paperswithcode.com/dataset/cnn-daily-mail-1

on establishing how well the ROUGE metric score evaluates the quality of a summary [119], both for the EATS and AATS techniques.

The goal of the second RQ, on the other hand, is to determine whether a single ATS algorithm execution produces better outcomes with respect to a multiple execution (in which we define *multiple execution* as a cascade execution of two ATS algorithms in which the output of the first becomes the input of the second), always using the ROUGE as a result evaluation metric. Briefly:

1. **RQ1:** How different is the ROUGE score for the EATS and AATS ? Can the quality of a summary produced by an ATS algorithm be accurately predicted by this metric score?
    a) *Object of study* is the ROUGE score determined in both the EATS and AATS algorithms;
    b) *Purpose* is to estimate the validity and effectiveness of this metric in both cases;
    c) *Perspective* is the viewpoint of a researcher;
    d) *Context* relating to the experimentation carried out is represented by the use of ATS algorithms on a standard dataset of texts and summaries.

2. **RQ2:** How does the multiple execution of a summary differ from the single execution (where the summary is produced by running a single algorithm)? Is it adequate to compare the two strategies using the ROUGE score?
    a) *Object of study* is the ROUGE score determined for the single and multiple ATS algorithms execution;
    b) *Purpose* is to evaluate the quality of the summaries produced by each of the two outlined approaches;
    c) *Perspective* is the viewpoint of a researcher;
    d) *Context* relating to the experimentation carried out is represented by the use of ATS algorithms on a standard dataset of texts and summaries.

4.4.3 *Experiment Planning*

The details of the experiment planning are illustrated in the section below. For the experiment to be successful, this step is essential because it helps to organize all the required tasks well and to monitor the process, defining how the experiment is conducted.

4.4.3.1 *Context Selection.*

The experimentation lasted several days and is generalized because it compares the validity and accuracy of the ROUGE metric (from the viewpoint of a researcher) for two different ATS approaches in the first RQ. Instead, the effectiveness of two different execution methods for the automatic development of summaries is evaluated in the second RQ. The ROUGE metric, which is used to judge all outcomes, ensures the comparability of the methodologies under consideration.

4.4.3.2 *Hypotheses Formulation.*

The testing of hypotheses serves as the foundation for the statistical analysis of an experiment. A hypothesis must be formally defined and the data collected during the course of the experiment is used to confirm or reject it. Two hypotheses have to be formulated: the *null* and the *alternative*. Below follows the formal description of both ones for the two RQs, taking into account the ROUGE metric for comparison.

1. **RQ1**
    a) *Null Hypothesis*: The AATS methods perform differently than the EATS approaches. (This is due to the fact that Extractive methods use portions of the original text in the output summary, which should produce a different overlap ratio of n-grams, as opposed to Abstractive methods, which use new terms in the generated summary and consequently different n-grams overlapping).

$$H_0 : \mu_{ROUGE\_Ext} \neq \mu_{ROUGE\_Abs} \tag{4.11}$$

where μ is the mean and ROUGE the score of each summary;

b) *Alternative Hypothesis*: The AATS approaches perform approximately as well as the EATS ones. (This would suggest that the ROUGE metric is inappropriate for the summary evaluation produced by the system).

$$H_A : \mu_{ROUGE\_Ext} = \mu_{ROUGE\_Abs} \tag{4.12}$$

where μ is the mean and ROUGE the score of each summary.

2. **RQ2**

a) *Null Hypothesis*: A multiple execution of ATS algorithms, compared to a single execution on the same input text, produces worse or the same results.

$$H_0 : \mu_{ROUGE\_Multiple} \leq \mu_{ROUGE\_Single} \tag{4.13}$$

where μ is the mean and ROUGE the score of each summary;

b) *Alternative Hypothesis*: A multiple execution of ATS algorithms, compared to a single execution on the same input text, produces better results.

$$H_A : \mu_{ROUGE\_Multiple} > \mu_{ROUGE\_Single} \tag{4.14}$$

where μ is the mean and ROUGE the score of each summary.

4.4.3.3 *Variable Selection.*

A key step during the planning of the experiment is the variable selection. They are divided into two groups:

- *independent variables* are those that can be managed and changed during the experiment;
- *dependent variables* are those that measure the impact of the experiment on the different combinations of the independent variables.

Consequently, the *independent variables* can have some effects on the *dependent variables*. In our case:

1. **RQ1**
    a) *Independent variables*: the EATS and AATS methodologies. For each of these, different algorithms will be performed;
    b) *Dependent Variables*: the ROUGE score for the outcome of each algorithm. The findings will be averaged to offer a single comparative measure.

2. **RQ2**
    a) *Independent variables*: ATS methodologies that include single and multiple executions of the summaries. For each of them, various algorithms will be used in different combinations;
    b) *Dependent Variables*: the ROUGE score for the output of each algorithm. The findings will be averaged to offer a single comparative measure.

4.4.3.4 *Subjects Selection.*

The selection of subjects is essential when performing an experiment since it affects how broadly the results may be generalized. To achieve this, an adequate sample size must be used to ensure that the whole population is represented in the selection. Probability or non-probability strategies can be used for sampling [134].

The *Simple Random Sampling* model is used in both the RQs, where subjects are randomly selected from a population list (in this experiment consisting of approximately 287.000 items). For computational efficiency, each algorithm is performed on a block of 1.000 of them. Each of these selected texts is processed through each algorithm, enabling a realistic comparison of the outcomes from the same input.

In the first trial, 40.000 summaries are produced, with 1.000 texts in each block. A total of 1.000 texts are taken into account in the second experiment due to the computational difficulty and the time required to complete the experiment.

4.4.3.5 *Design Type Choice.*

Since an experiment is a sequence of tests, in order to conduct it well, the series of experiments must be carefully designed. This is done by choosing the design type of the experiment, which specifies how the tests are organized and conducted. Below there is a description of our test methodologies.

- **Principle General Design**: We have chosen to use balancing and randomization techniques, so all the tests are conducted with randomized chunks of texts. The order in which the independent variables are selected is also at random, but this has no effect on the outcomes of the experiment. Each test for the *balancing design principle* [137] was done on chunks of 1.000 texts to evaluate because this allows very good results and statistically valid conclusions.
- **Standard Design Type**: As Design Type for RQ1, a factor with two treatments was chosen. Indeed, we aim to compare the EATS and AATS techniques through these experiments. The same Design Type is used for RQ2 too. We are particularly interested in comparing the performance of a single versus a multiple execution of summaries. All of the algorithms that have been considered are used in each experiment. In particular, we examine the execution of an EATS algorithm followed by the execution of an AATS algorithm for multiple tasks (and vice versa).

4.4.3.6 *Tools.*

The Python programming language was used to create the complete software needed to run this experiment. The findings have been acquired and accurately analyzed due to the implementation of summarization procedures in addition to the implementation of the ATS algorithms.

4.4.4 *Operation Phase*

The execution of a designed and planned experiment allows to collect all the information necessary for the analysis of the

results. It is precisely in this phase that the foreseen treatments are applied to the experimental subjects. Three steps make up this operation part of an experiment: preparation, execution, and data validation.

4.4.4.1 *Preparation.*

There are various aspects that need to be addressed before the experiment was conducted. Our role during this stage was to set up the environment where the experiment would take place. We specifically verified the functionality of the code used to extract the random texts from the dataset, the accuracy of the algorithms, and the results the metric score produced. It was important to set up the code for collecting the information. In this situation, all computed scores have been included in a dataset that also contained all average outcomes for each block of summaries. This was done for each algorithm used in the experiment.

4.4.4.2 *Execution.*

The processing for this experiment took several days due to the time the algorithms needed to calculate all of the results from the several tests that were performed. Indeed, all of the subjects (the texts) were chosen from the input dataset and then sent as input to the various algorithms for the generation of summaries, for each of them. In particular, it was used a *CSV* file containing all the summaries. It was organized into three columns: one for the original text, one for the gold standard, and one for the related summary for each algorithm. Each line of the file represented a different text randomly chosen from the dataset. Following this phase, all the summaries were properly evaluated with the use of the ROUGE metric, which allowed their final analysis.

4.4.4.3 *Data validation.*

Data validation was done by selecting entries at random, looking over them, and checking the consistency of the CSV files. Testing was done on the ROUGE scores of every sample to verify if they satisfied the criterion established by the designers of the various algorithms.

4.4.5 *Results Analysis*

In this part, the findings from the two experiments are discussed, evaluated, and interpreted, with some graphs emphasizing their statistical validity.

4.4.5.1 *Descriptive Analysis.*

A few important aspects of the results obtained are presented, taking into account the huge number of tests conducted and assuming that each block of summaries has the same distributional shape. This is guaranteed by the fact that subjects are chosen at random from the general community.

A random execution of the *textRank* algorithm is investigated for this purpose. First of all, some indicators of the centrality of the data are provided in Table 4.1: the mean, the median and the standard deviation. Subsequently, a boxplot and a histogram of randomly generated results were used to analyze the distribution of the obtained results. It is important to underline that each result refers to a collection of 1.000 summaries that differ depending on the algorithm and input text. In particular, Figure 4.25 shows the 1.000 scores for ROUGE-1, ROUGE-2, and ROUGE-L obtained from the *textRank* algorithm execution. Some interesting trends can be observed:

- as we could expect, ROUGE-2 has smaller values than ROUGE-1 and ROUGE-L;
- it is possible to find a discrete number of outliers, particularly on the top of the boxplot. This holds true for each of the three types of ROUGE;

Table 4.1: Descriptive statistics.

ROUGE Metric	Mean	Median	St Dev
ROUGE-1	0.205	0.194	0.002
ROUGE-2	0.059	0.041	0.002
ROUGE-L	0.204	0.189	0.003

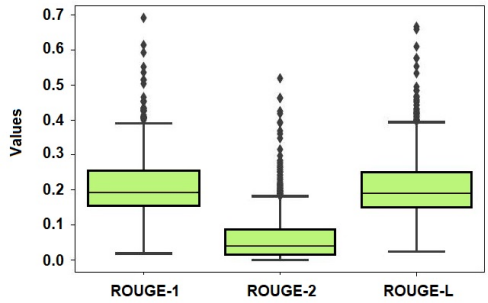


Figure 4.25: Boxplot of ROUGE metric scores computed on 1.000 summaries by the textRank algorithm.

- the borders of the box are divided nearly equally by the median line in ROUGE-1 and ROUGE-L. This indicates that, particularly in the interquartile range, the data are very well-centered;
- a positive skewness in the ROUGE-2 metric can be observed. This may be detected by looking at the median line close to the bottom edge of the box, where the whisker is shorter;
- given that the whisker and the edges of the corresponding boxes are symmetric, a normal distribution may be hypothesized for the ROUGE-1 and ROUGE-L measures.

As regards the visible outliers, it is easy to show the number of data points that are under or above Q1 and Q3. We have:

- ROUGE-1: 36 observations (3.6% of the data);
- ROUGE-2: 67 observations (6.7% of the data);
- ROUGE-l: 41 observations (4.1% of the data).

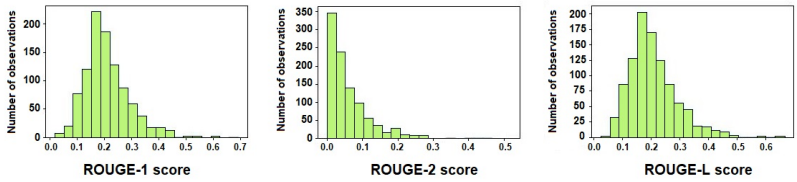


Figure 4.26: Histogram showing the data distribution for ROUGE-1, ROUGE-2, ROUGE-L scores using the textRank algorithm.

Table 4.2: Mean and Standard Deviation for all the algorithms and ROUGE metrics used.

Algorithm	**ROUGE-1**		**ROUGE-2**		**ROUGE-L**	
Extractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
textRank	0.205	0.002	0.059	0.002	0.204	0.003
lsa	0.223	0.004	0.056	0.003	0.205	0.004
luhn	0.220	0.003	0.066	0.002	0.220	0.003
lexRank	0.242	0.003	0.071	0.002	0.232	0.003
Abstractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
word2vec	0.213	0.003	0.058	0.002	0.205	0.003
doc2vec	0.215	0.002	0.059	0.002	0.206	0.002
glove	0.213	0.003	0.058	0.002	0.205	0.003

Avoiding considering ROUGE-2, which is the most skewed distribution, both ROUGE-1 and ROUGE-L are under 5%.

Finally, Figure 4.26 shows each ROUGE measure distribution by three representative histograms. As anticipated by the precedent boxplot, ROUGE-1 and ROUGE-L approximate quite well the normal distribution. This guarantees the good distribution of data points along all the observations and allows us to consider the mean as a valid representation measure for them.

4.4.5.2 *RQ1 results.*

The first RQ concerned determining the applicability of the ROUGE metric in evaluating ATS algorithms.

In particular, the experiment design guidelines included comparing the outcomes of both the EATS and AATS techniques using a random sample of texts. In Table 4.2 are resumed *mean* and *standard deviation*, for each of the three ROUGE measures in relation to each of the algorithms used in this experiment.

For the experiment, 40 blocks with 1.000 summaries each were analyzed. An average within each block and an overall average for each algorithm were calculated to illustrate the outcomes. The average score for the seven algorithms tested with respect to the three ROUGE measures is shown in Figure 4.27. The first four algorithms (*textRank*, *lsa*, *luhn*, and *lexRank*) are Extractive,

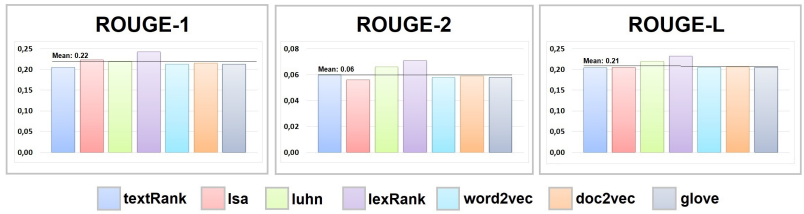


Figure 4.27: ROUGE average scores of the experiment conducted on EATS and AATS algorithms.

whereas the latter three are Abstractive (*glove*, *word2vec*, and *doc2vec*).

The mean of each of the algorithms is quite similar. In this case, *LexRank* results to be the most efficient algorithm, scoring about 10% more than the others. On the other hand, the values for the Abstractive approaches have extremely comparable values, and even if only slightly, all of their scores are below the mean.

To confirm or reject the hypothesis, a statistical validity test was also conducted. This was executed by doing a *t-test* [121] on the distribution of results for each summary, which were grouped for EATS and AATS. The 40.000 summaries that were examined for this test have the same freedom degrees. The *p-value* [136] for the statistical validity of the experiment is resulted to be 2.2e-16, which is less than the required 0.05 and this supports the alternative hypothesis that the EATS and AATS ROUGE scores are equivalent. The initial assumption was that EATS methods would work much better than the AATS ones. Instead, the outcomes showed that this assumption is not true. In fact, both algorithms obtained very similar results in most of the cases. This serves to emphasize the fact that, for the intrinsic way in which it is used for evaluation, ROUGE is not the best metric to use when assessing ATS algorithms.

Some considerations can be made on these results. In fact, ROUGE compares a summary produced by a machine with one written by a human, and the score is based on a statistical analysis of the number of n-grams that overlap in the two texts. Following this logic, the more summaries use different words, the more poorly the ROUGE metric will perform. In that case, there will be a low ratio of overlap of n-grams. The weak points

of this reasoning are different. A metric like this does not take care of the semantic meaning of the sentences. But a topic can be expressed in different ways and with completely different words. In that case, the ROUGE score can be misleading in the evaluation of the quality of the summary, mostly if the focus is on the coverage of the source key topics. As a result, Abstractive methods should suffer greatly, and it is possible to come to the conclusion that algorithms that extract random words from the source texts can perform in some cases very well. Maybe a lucky case where the overlap of n-grams matches perfectly.

Furthermore, to verify the conclusion that ROUGE is not particularly representative, we can also look at the *gold standard*, which is the human-generated summary and which should be the best available summary of an input text (it is the target summary for the ATS algorithms). If we consider different summaries generated by humans, starting from the same source text, the results can be widely different but be all valid and acceptable. For sure we can achieve excellent readability and a sure good syntactic composition of the sentences. Despite this, if we calculate the ROUGE score between two gold standards, we would not appreciate it. Of course, each human has the ability to create their own summary using different words, different compositions of the text and a strong preference for one subject over another. ROUGE does not take care of all this into account and it produces outcomes that do not accurately reflect the essential elements of a summary.

The findings largely support our alternative hypothesis for the initial study issue. According to the ROUGE score, the AATS algorithms perform quite similarly to the EATS ones. This confirms that the ROUGE metric is a not very valid metric for assessing the quality of the summaries produced by ATS algorithms.

4.4.5.3 *RQ2 results.*

The second RQ had the goal of comparing the effect of a single execution versus a multiple execution of an ATS algorithm. Two options for multiple executions were considered:

1. An *Extractive* algorithm on the input of the *Abstractive* ones;
2. An *Abstractive* algorithm on the input of the *Extractive* ones.

Table 4.3 shows the achieved results for each block of summaries. In this table are visible in the first two columns the frequencies with which a single execution outperforms a multiple one and vice versa, relating this value in percentage to the total number of samples. Instead, in the last two columns are shown the average ROUGE scores for the two different execution methodologies (the multiple execution was performed taking into account each combination of *Extractive-Abstractive* algorithms, and the general average by typology was then determined).

These outcomes are represented graphically in Figure 4.28. On the left are shown the results relating to the average of the ROUGE-1 score (only this one has been included for ease of display) obtained in case of execution of each of the *EATS* algorithms, compared by the execution of the same algorithm on the output produced by an *AATS* one. Instead, the specular result taking the *AATS* methods into account appears on the right. In this visualization, it is evident that in the majority of the tested cases, multiple executions (in red) have performed better than single ones (in blue).

Table 4.3: Comparison between a single and a multiple execution of Extractive (resp Abstractive) algorithms on the input of Abstractive (resp Extractive) ones.

	Performances		Mean	
Extractive Algorithm	**Single execution**	**Multi-exec (on Abs)**	**Single execution**	**Multi-exec (on Abs)**
textRank	34.88%	65.12%	0.2100	0.2396
lsa	43.95%	56.05%	0.2235	0.2399
luhn	41.65%	58.35%	0.2241	0.2396
lexrank	52.50%	47.50%	0.2566	0.2424
Abstractive Algorithm	**Single execution**	**Multi-exec (on Ext)**	**Single execution**	**Multi-exec (on Ext)**
word2vec	39.00%	61.00%	0.2145	0.2391
doc2vec	41.16%	58.84%	0.2177	0.2388
glove	38.60%	61.40%	0.2133	0.2389

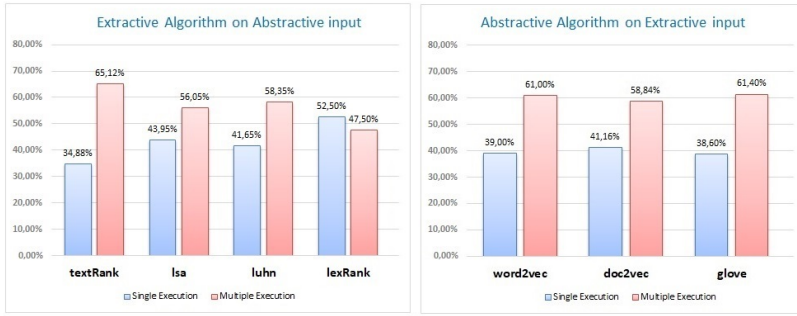


Figure 4.28: Results comparison for summaries between a single execution (blue) and a multiple execution (red) for each type of methodology.

Also in this case is performed the t-test in order to ensure the statistical validity of the experiment. This was conducted considering the differences between the Extractive (on Abstractive) and Abstractive (on Extractive) methodologies. For each test was compared the multiple execution approach against the single one on a population composed of 1.000 paired summaries.

The findings are quite different: the t-test for the Extractive (on Abstractive) approach yielded a p-value of 0.4, indicating that this experiment lacks statistical validity. Instead, the alternative hypothesis that multiple executions outperform single executions is confirmed by the t-test for the Abstractive (on Extractive) method, which produced a p-value of 0.018, which is lower than the necessary 0.05 for statistical validity.

These results are significant since they demonstrate that in almost all algorithms, the multiple execution strategy outperformed the single one. One possible reason is the compression ratio obtained from multiple algorithm executions where a first iteration can remove unnecessary information, whilst a second one compresses key concepts into a summary that scores higher. This indicates how the compression ratio has caused algorithms to save as much information from the source text as possible to include it in the output summary. So we may think that AATS algorithms, in this case, can act like the EATS ones.

Since the ratio of n-grams overlapping might lead to misleading results, having a more compressed reference summary can benefit from the ROUGE score. In fact, despite the ROUGE score

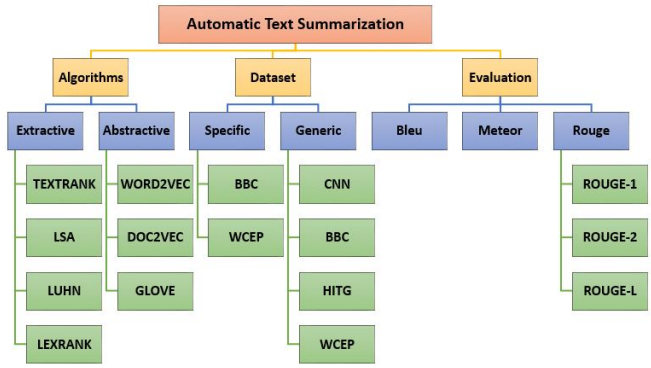


Figure 4.29: Research study framework.

being very good for multiple executions, should be seriously taken into consideration the readability of the summary. The results of a better ROUGE score can sometimes bring unconnected and semantically complex sentences and words in the output summary.

4.5 FURTHER INVESTIGATION

The diagram in Figure 4.29 shows the framework of this further investigation and lists the various ATS algorithms involved (the same of the first experiment), the datasets utilized to produce the final outcomes, and the evaluation metrics that we compared in the experiments.

4.5.1 *Dataset*

All of the experiments were done on four different datasets:

- **CNN/Daily Mail** dataset. It is consisting of CNN and Daily Mail articles and editorial news is the most used reference dataset in the ATS field [147]. The original corpus was modified to include multi-sentence summaries [88]. This database today has two distinct versions. In the first [151] entities are replaced with anonymous ones whilst in the second [120] all them are contained;

- **BBC News**. It is frequently used for EATS and includes papers from the BBC News website that correspond to news from 2004 to 2005 in five different theme areas [37];
- **HITG** dataset of summaries[4]. It has over 100.000 texts, each of which comprises a variety of information in addition to the articles and a short description[5];
- **WCEP** is a multi-document summary dataset from the Current Events Portal of Wikipedia [35]. Each summary includes brief, human-written descriptions of recent news events that are each paired with a selection of news items connected to the event[6].

4.5.2 *Research Questions*

This further investigation tends to refine the results of the first one. In particular, it tends to answer to other two specific RQs.

- The *first one* aims to expand on what was learned in the previous experiment, by determining whether the same findings were obtained when the field of interest of the source datasets was narrowed;
- The *second one*, then, compares many metrics, including some of the most popular in the ATS field, to see if one of them can be more effective than the others in regards to evaluating ATS algorithms.

Briefly:

1. **RQ3:** How different is the ROUGE score obtained by the EATS approaches compared to the AATS ones, when it is restricted the starting dataset for the different algorithms to a narrow interest field against a general interest field?

4 https://www.kaggle.com/sunnysai12345/news-summary
5 The only news content that was scraped was from the Hindu, Indian Times, and Guardian, as well as the news summaries from Inshorts. February to August of 2017 is the time frame.
6 These articles are composed of content automatically retrieved from the Common Crawl News collection and sources cited by WCEP editors.

a) *Object of study* is the ROUGE score determined for each dataset among all algorithms;

b) *Purpose* is related to confirming the weakness and inefficacy of this metric for such restrictions;

c) *Perspective* is the viewpoint of a researcher;

d) *Context* relating to the experimentation carried out is represented by the use of ATS algorithms on a standard dataset of texts and summaries.

2. **RQ4:** Can we clearly distinguish between the EATS and AATS techniques using the different evaluation metrics BLEU, METEOR, and ROUGE to compare the various algorithms?

   a) *Object of study* is represented by the various results of the three metrics for the different datasets considered;

   b) *Purpose* is to show that none of these metrics can effectively assess the different EATS and AATS techniques;

   c) *Perspective* is the viewpoint of a researcher;

   d) *Context* relating to the experimentation carried out is represented by the use of ATS algorithms on a standard dataset of texts and summaries.

4.5.3 *Experiment Planning*

Planning is the first step in the experimental process. In order to highlight the inefficiency of the ROUGE metric when the field of interest of a dataset is restricted to a particular context, the first experiment compares the validity and accuracy of this metric (from the viewpoint of a researcher) on two types of approaches, the EATS and the AATS ones. Instead, the goal of the second one is to compare the findings with two other metrics that are often used in the literature to evaluate ATS approaches.

The selection of subjects is fundamental when conducting experiments since it has a direct impact on how extensively the findings may be generalized. To this end, sampling must represent the whole population and, as a result, both probabilistic and non-probabilistic methodologies can be used to determine the sample size [134].

Both of the RQs employ the *Simple Random Sampling* model, in which subjects are randomly selected from a population list. Each algorithm is executed on different blocks of them for computational efficiency (the dimension of each block depends on the size of the original dataset). Every one of these chosen texts is processed using a distinct algorithm and set of metrics, allowing for a fair comparison of the results obtained from the same input. For the RQ3 and all general topics, we looked at two different datasets separated into various topics. We have chosen:

- for the **BBC** dataset we have considered 300 blocks of 200 text each one;
- for the **WCEP** dataset we have considered 300 blocks of 120 text each.

For RQ4, on the other hand, we have chosen:

- for the **BBC** dataset we have considered 300 blocks of 200 text each one;
- for the **CNN** dataset we have considered 300 blocks of 200 text each one;
- for the **HITG** dataset we have considered 300 blocks of 1000 text each one;
- for the **WCEP** dataset we have considered 300 blocks of 120 text each.

The size of the datasets, the difficulty of the computing, and the amount of time needed to complete the experiment all are factors that have influenced the different chunk sizes. The majority of the tests took many days to complete. As regards the *Design Type Choice* and *Tools* they are analogous to those of the previous investigation.

4.5.4 *Operation Phase*

The experiment operation phase consists of three steps:

4.5.4.1 *Preparation.*

During this step, it was crucial to confirm the accuracy of the code that extracts the random texts from the various datasets used for each experiment, as well as the outcomes of the ROUGE, BLEU, and METEOR metrics. A fundamental check has also been done on the code that acquires and evaluates the results. In fact, a mistake in precision could undermine the outcomes. Each experimental result which included the scores for each kind of metric was subdivided into different datasets.

4.5.4.2 *Execution.*

The experiment ran for many days because it took a lot of computation time to test different databases using the three specific metrics ROUGE, BLEU, and METEOR. The algorithms for the two RQs were performed grouped by the specific ATS techniques and evaluation metrics and based on the same input documents. For both RQs, the dataset texts were initially chosen at random and summarized using the seven different algorithms. All the results were grouped in *CSV* files.

4.5.4.3 *Data validation.*

Data validation was done by randomly checking the items from each of the CSV files to make sure they were consistent. Additional testing was conducted, analyzing all of the data, to see if the scores of the various samples for each of the metrics involved satisfied the quality standards of the algorithm developers.

4.5.5 *Results Analysis*

The results of the two experiments are explained, evaluated, and interpreted in this section, with the help of some graphs and tables.

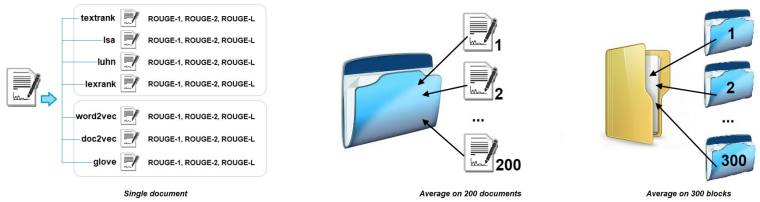


Figure 4.30: Experiment schema on the *BBC* dataset.

4.5.5.1 *RQ3 results.*

In this experiment, the *BBC* and the *WCEP* datasets were considered because they are composed of documents related to different topics.

As highlighted above, the *BBC* dataset is made up of texts that are broken up into subcategories where each subpart concerns topics of the same nature. To better understand what the final summary of the results will be, let us consider the *'Politics'* topic. For this subpart of the dataset, the experimentation was carried out on 300 blocks, each consisting of 200 texts taken randomly from those relating to this topic.

The three ROUGE scores were calculated for each text. The average of these ratings was then determined for each block of

Table 4.4: Comparison of ROUGE measures scores for the BBC dataset for the *'**Politics**'* topic.

algorithm	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
textrank	0,318	0,237	0,314
lsa	0,216	0,124	0,209
luhn	0,334	0,254	0,328
lexrank	0,294	0,215	0,290
word2vec	0,277	0,190	0,270
doc2vec	0,285	0,199	0,279
glove	0,292	0,204	0,286
Extractive	0,290	0,207	0,285
Abstractive	0,285	0,198	0,278
Difference	0,005	0,009	0,007

Table 4.5: Rouge results for BBC dataset.

		BUSINESS	ENTERT.	POLITICS	SPORT	TECH	ALL
Rouge-1	Extractive	0,268	0,282	0,290	0,369	0,265	0,298
	Abstractive	0,275	0,283	0,285	0,339	0,259	0,291
	Differences	0,007	0,001	0,005	0,030	0,006	0,007
Rouge-2	Extractive	0,192	0,207	0,207	0,280	0,181	0,217
	Abstractive	0,194	0,204	0,198	0,239	0,172	0,203
	Differences	0,002	0,003	0,009	0,041	0,009	0,014
Rouge-l	Extractive	0,264	0,277	0,285	0,363	0,259	0,293
	Abstractive	0,271	0,279	0,278	0,331	0,253	0,285
	Differences	0,007	0,002	0,007	0,032	0,006	0,008

200 texts. Finally, the general mean of the scores was calculated for the 300 considered blocks (see Figure 4.30).

The overall findings of this experiment for the *'Politics'* topic are displayed in Table 4.4, where the two evaluations for the different EATS and AATS algorithm types have been grouped. Finally, the difference in scores between them is determined. This table represents a detailed extension of one of the columns related to the whole outcome.

These results were found for the *BBC* and *WCEP* datasets, respectively, and are reported in the tables 4.5 and Table4.6. These tables show the findings of tests performed on texts with a single topic (the initial columns), as well as tests performed

Table 4.6: Rouge results for WCEP dataset.

		ART	BUSINESS	DISASTERS	...	WAR	ALL
Rouge-1	Extractive	0,235	0,224	0,243	...	0,259	0,241
	Abstractive	0,232	0,232	0,238	...	0,254	0,238
	Differences	0,003	0,008	0,005	...	0,005	0,003
Rouge-2	Extractive	0,053	0,045	0,055	...	0,058	0,053
	Abstractive	0,049	0,044	0,050	...	0,053	0,049
	Differences	0,004	0,001	0,005	...	0,005	0,004
Rouge-l	Extractive	0,194	0,186	0,198	...	0,209	0,196
	Abstractive	0,191	0,192	0,192	...	0,205	0,192
	Differences	0,003	0,006	0,006	...	0,004	0,004

on texts chosen at random from the full dataset, meaning texts without a specified topic (last column).

The results demonstrate that, even when the datasets are restricted to topics within the same field of interest (first columns of the tables), the algorithmic scores are quite similar to those obtained when the topics are not distinct (last column of the tables). Consequently, it can be seen how the findings from the first stage of the research are supported and are therefore not dependent on the restriction of the datasets to particular interest fields.

Additionally, the scores of the EATS and AATS methodologies, on average, are not very different from one another. As a result, ROUGE turns out to be not effective for the evaluation of the goodness of the ATS algorithms in this case too.

4.5.5.2 *RQ4 results.*

In the second experiment, four large datasets were used to evaluate three different metrics. The objective was to determine the average of the scores of these metrics for each algorithm under consideration, in order to determine whether one of them could distinguish between EATS and AATS approaches. Before discussing the overall results on all the datasets taken into consideration, we will only go into detail about each of them. The findings for all the datasets under consideration were obtained following the same scoring methodology as in the RQ3.

Table 4.7: Comparison of mean scores of metrics for the BBC dataset.

algorithm	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
textrank	0,116	0,242	0,318	0,238	0,314
lsa	0,061	0,168	0,238	0,147	0,232
luhn	0,139	0,261	0,338	0,261	0,333
lexrank	0,096	0,219	0,298	0,221	0,294
word2vec	0,089	0,217	0,289	0,202	0,283
doc2vec	0,089	0,216	0,288	0,201	0,282
glove	0,092	0,222	0,296	0,207	0,290
Extractive	0,103	0,223	0,298	0,217	0,293
Abstractive	0,090	0,218	0,291	0,203	0,285
Difference	0,013	0,005	0,007	0,014	0,008

BBC dataset

In Table 4.7 are shown the results approximated to the third decimal place of all algorithms for each metric, as well as the differences between the two average scores for the EATS and AATS category of algorithms for the *BBC* dataset.

Instead, a graphical representation of all the scores from the different algorithms for the BLEU, METEOR, and ROUGE metrics is given in Figure 4.31, with a red line indicating the average score and the scores themselves being represented by bars.

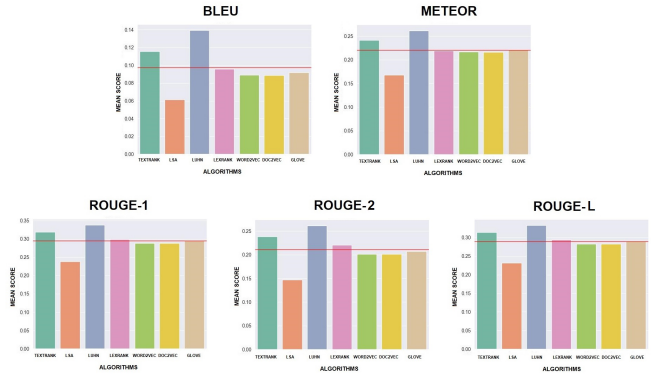


Figure 4.31: Mean average BLEU, METEOR and ROUGE-1 scores BBC dataset for EATS and AATS algorithms.

Table 4.8: Comparison of mean scores of metrics for the CNN Daily-mail dataset.

algorithm	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
textrank	0,040	0,267	0,400	0,136	0,365
lsa	0,040	0,243	0,356	0,111	0,326
luhn	0,050	0,293	0,438	0,169	0,401
lexrank	0,049	0,279	0,414	0,151	0,378
word2vec	0,043	0,251	0,354	0,116	0,320
doc2vec	0,043	0,254	0,358	0,118	0,324
glove	0,042	0,253	0,358	0,117	0,323
Extractive	0,045	0,271	0,402	0,142	0,368
Abstractive	0,043	0,253	0,357	0,117	0,322
Difference	0,002	0,018	0,045	0,025	0,046

CNN Daily Mail dataset

In Table 4.8 are shown the results approximated to the third decimal place of all algorithms for each metric, as well as the differences between the two average scores for the EATS and AATS category of algorithms for the *CNN Daily Mail* dataset.

Instead, a graphical representation of all the scores from the different algorithms for the BLEU, METEOR, and ROUGE metrics is given in Figure 4.32, with a red line indicating the average score and the scores themselves being represented by bars.

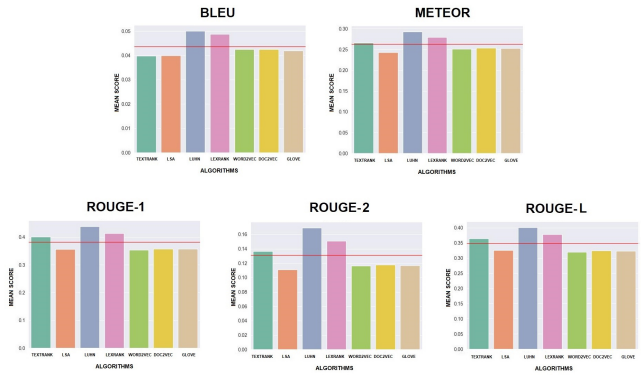


Figure 4.32: Mean average BLEU, METEOR and ROUGE-1 scores CNN Daily Mail dataset for EATS and AATS algorithms.

Table 4.9: Comparison of mean scores of metrics for the HITG dataset.

algorithm	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
textrank	0,014	0,274	0,361	0,126	0,314
lsa	0,014	0,269	0,352	0,124	0,306
luhn	0,013	0,254	0,342	0,116	0,297
lexrank	0,027	0,427	0,541	0,223	0,471
word2vec	0,016	0,309	0,407	0,148	0,352
doc2vec	0,014	0,288	0,382	0,135	0,331
glove	0,016	0,300	0,397	0,142	0,344
Extractive	0,017	0,306	0,399	0,147	0,347
Abstractive	0,015	0,299	0,395	0,142	0,342
Difference	0,002	0,007	0,004	0,005	0,005

HITG dataset

In Table 4.9 are shown the results approximated to the third decimal place of all algorithms for each metric, as well as the differences between the two average scores for the EATS and AATS category of algorithms for the *HITG* dataset.

Instead, a graphical representation of all the scores from the different algorithms for the BLEU, METEOR, and ROUGE metrics is given in Figure 4.33, with a red line indicating the average score and the scores themselves being represented by bars.

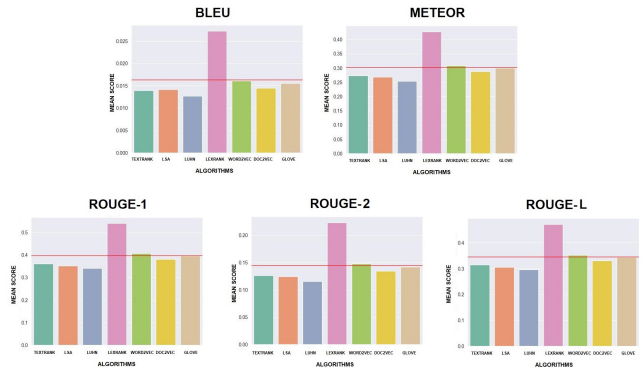


Figure 4.33: Mean average BLEU, METEOR and ROUGE-1 scores HITG dataset for EATS and AATS algorithms.

Table 4.10: Comparison of mean scores of metrics for the WCEP dataset.

algorithm	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
textrank	0,011	0,177	0,253	0,057	0,206
lsa	0,006	0,131	0,192	0,034	0,156
luhn	0,011	0,182	0,265	0,061	0,215
lexrank	0,012	0,179	0,252	0,060	0,206
word2vec	0,008	0,163	0,236	0,049	0,190
doc2vec	0,008	0,164	0,238	0,048	0,193
glove	0,008	0,166	0,241	0,049	0,194
Extractive	0,010	0,167	0,241	0,053	0,196
Abstractive	0,008	0,164	0,238	0,049	0,192
Difference	0,002	0,003	0,003	0,004	0,004

WCEP dataset

In Table 4.10 are shown the results approximated to the third decimal place of all algorithms for each metric, as well as the differences between the two average scores for the EATS and AATS category of algorithms for the *WCEP* dataset.

Instead, a graphical representation of all the scores from the different algorithms for the BLEU, METEOR, and ROUGE metrics is given in Figure 4.34, with a red line indicating the average score and the scores themselves being represented by bars.

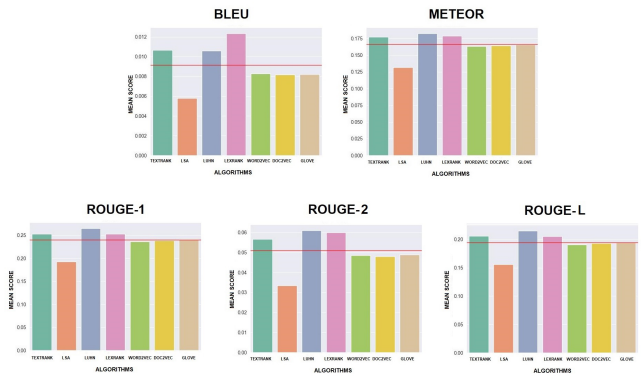


Figure 4.34: Mean average BLEU, METEOR and ROUGE-1 scores WCEP dataset for EATS and AATS algorithms.

Table 4.11: Results summary for all datasets.

		BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L
BBC	*Ext*	0,103	0,223	0,298	0,217	0,293
	Abs	0,090	0,218	0,291	0,203	0,285
	Diff	0,013	0,005	0,007	0,014	0,008
CNN	*Ext*	0,045	0,271	0,402	0,142	0,368
	Abs	0,043	0,253	0,357	0,117	0,322
	Diff	0,002	0,018	0,045	0,025	0,046
HITG	*Ext*	0,017	0,306	0,399	0,147	0,347
	Abs	0,015	0,299	0,395	0,142	0,342
	Diff	0,002	0,007	0,004	0,005	0,005
WCEP	*Ext*	0,010	0,167	0,241	0,053	0,196
	Abs	0,008	0,164	0,238	0,049	0,192
	Diff	0,002	0,003	0,003	0,004	0,004

Table 4.11 resumes all the findings for the various datasets. It is evident that these metrics fail to distinguish between the EATS and AATS approaches because the average scores are quite similar among all of them, in the order of hundredths of the unit. This proves that none of these three metrics is useful to evaluate the goodness of ATS algorithms.

4.6 VALIDITY EVALUATION AND THREATS DISCUSSION

A big issue with the plausibility of the experiment and the reproducibility of its results is the validity of the outcomes. There are different classification schemes for different types of threats to the validity of an experiment. *Conclusion*, *internal*, *construct* and *external* validity are the four threats. A discussion of them in relation to our first RQ is provided below (but it is extendable to the other three RQs).

4.6.1 *Conclusion Validity*

The threat of having *low statistical power* was excluded. In fact, all the experiments have been completed and the collected results

are based on solid scientific and statistical security. Furthermore, the metric used to compare the two methods performed in the experiment returns a well-defined numerical score, which is well comparable and can be analyzed without losing validity.

The dataset used can be influenced by the *hypothesis of violation of the statistical tests*. Indeed, based for example on the syntax and semantics of the texts used as input, there may be some variations in the scores obtained through the metric used. However, having performed a lot of tests for each of the algorithms, and having calculated the general average of all the scores obtained (1000 summaries chosen at random) this threat really has a negligible impact.

The number of tests conducted for the research was only performed for statistical analysis, hence *Fishing* is not included.

Regarding the *reliability of the measurements*, the accuracy of the results is without debate. In practice, all ROUGE metrics are calculated using a common Python library, and since the algorithm is well-defined, the outcomes are therefore repeatable.

The subject selection and design type were combined with a standard execution approach to ensure the *reliability of treatment implementation* and to prevent incorrect execution and the threat it presents.

Random irrelevances in the experimental setting are not to be taken into account because the performance is completed in a controlled environment without the potential of any intervention from outside events. The *random heterogeneity of subjects* is also minimized by averaging the outcomes of the execution for 1000 texts.

4.6.2 *Internal Validity*

There are several fundamental aspects concerning the analysis of internal validity, which will be examined one by one. *Historical threats* are prevented since there is no chance that the outcome will change if the same experiment is repeated using the same algorithm at a different time interval because the experiment is not dependent on the advancement of time.

The same is true for *maturation risks*, as findings will be kept stable as time goes on because algorithms do not preserve information over time or across different experiment executions.

There are no presuppositions for the *testing threats*, as an algorithm does not have the difficulty of a human giving varied answers during the test owing to some knowledge of the technique, therefore results are similar across all tests.

Instrumentation threats may result in problems. Particularly, the use of external libraries that include the algorithms employed during the experiment may have various bugs, such as implementation faults or errors. Furthermore, both the computing of the scores of the summaries and the storage of Rouge scores in the produced software datasets can contain inaccuracies, particularly during the implementation process. In order to address these kinds of issue, a thorough analysis of the package documentation for the software in use was done, and the standard performance of each method was compared to the average standard quality offered by the creators.

The validity evaluation step does not take into account human behaviours such as *Statistical regression*, selection, and death risks.

4.6.3 *Construct Validity*

Construct validity refers to how well the findings of an experiment may be applied to the concept or theory. A construct may not be properly described, as suggested by the *Inadequate preoperative explanation of constructs*. For instance, saying *"one is better than the other"* can mean a number of different things because the word *"better"* is not clearly defined. But in the case of this experimentation, all the used metrics allow a perfect mathematical comparison between numerical values.

The *mono-method bias* bias suggests that using just one type of measure or set of observations increases the probability that the experiment will be misleading if this measurement introduces a measurement bias[7]. Using a range of measurements and having them cross-checked amongst each other could be a solution.

7 "Measurement bias" refers to any systematic or non-random error that occurs in a study data collection. Another generic term for this type of bias is "detection bias"

However, using a second metric is not a possible option because we are only evaluating ROUGE as a single metric for the quality of text summaries. Instead, in the last experiment, we dealt with this issue and were able to show that the results were valid by comparing different metrics.

Concerning the aspect related to *Confounding constructs and Construct Levels* it is important to underline that there are occasions when the difficulties are not so much the existence or absence of the construct but rather the levels that it assumes. In the whole research, in all of the RQs taken into account this problem is dealt with using various ATS algorithms for each research investigation, in order to have a valid statistical meaning and the levels of each algorithm correspond to those of the construct. Finally, the final score is calculated by averaging all methods in order to compare the two metrics.

Threats that are closely related to human behaviour in an experiment include *Interaction of different treatments and between tests and treatments*. For the tests conducted for this investigation, this is excluded. Other risks in this group have not been taken into consideration because of how closely they connect to the subject behaviour of an experiment.

4.6.4 *External Validity*

External validity threats are conditions that limit the ability to generalize the experiment results. The *interaction of selections and treatments* refers to the effect of having a non-representative sample of the population to generalize. For this work, this type of threat is very important because the used datasets refer to generic texts to be summarized. But the chosen sample size is composed of numerous blocks each with hundreds of texts. As a result, for each dataset, the total sample size is shown to be significantly more than what is required in the literature to accurately reflect the population.

Lack of a representative experimental environment or material is referred to as the *interaction of settings and treatments*. This might be connected to the test datasets and methods for the planned research. The used algorithms are not perfect because there is not enough processing capacity to conduct the experiment with

more complex approaches. This is why different algorithms were used to be able to average their results, generalize the outcome as much as possible, and enable other researchers to experiment in the same environment. Instead, the datasets considered are the ones that have been used the most to evaluate the efficacy of ATS algorithms in the literature.

The only threat to the outcomes of the experiments from the perspective of the *"interaction of history and therapy"* is the possibility of publication of new, more powerful ATS methods or updated versions of the datasets that were examined.

4.7 CONCLUSIONS

The most suitable approach in the literature for comparing a system-generated summary to one made by a human is the ROUGE metric. The final score is determined by counting how many n-grams overlap between the two texts. The major goal of this research was to examine how this metric works in relation to the task it is intended to complete, to discover whether it was efficient in judging the superiority of one ATS algorithm summary over another. In particular, it became clear from the first experiment that ROUGE was not effective for the ATS tasks. Additionally, we have discovered that multiple executions give superior results versus a single execution (even when evaluated with ROUGE).

We furthered our investigation based on these first findings to determine whether they persisted even when the field of interest of a dataset was narrowed. In fact, because of how this assessment metric based on n-grams is constructed, EATS algorithms developed utilizing sections of the actual text should perform better than AATS ones. Even then, research suggests that ROUGE is inefficient for determining the quality of a summary because it gives results that are remarkably comparable for both methods. In consideration of the readability and grammatical accuracy factors, a high ROUGE score does not necessarily indicate a high quality of the summary. Finally, to determine if there was a more efficient metric for evaluating the automatically generated summaries, we evaluated the results obtained using two additional evaluation metrics that were among the most used, on various

datasets. It has been demonstrated through experimentation that there is currently no effective metric for assessing auto-generated summaries, showing that this is still an unexplored area of study.

5 CONCLUSIONS

This chapter concludes the thesis. In particular, Section 5.1 summarizes the completed work during this research path, discussing the different contributions made. Finally, Section 5.2 offers some guidelines and suggestions for additional tasks that can be performed, starting from what has already been achieved.

5.1 SUMMARY

In this thesis, we concentrated on two important AI applications respectively on structured and unstructured data. The first is about the DI, and the second is on the ATS, specifically the evaluation of the algorithms that generate the summaries. We started out by specifically looking at the many approaches and techniques that are currently used whilst looking into the challenges that are still open in both fields.

Consequence of the expansion of the internet today is the explosion in data generation (defined "*infobesity*"). Online data production is estimated to exceed 3 quintillion bytes each day. This unit of measurement was specifically created to give an idea of the immense amount of data known as "BigData". So, many companies today invest in BigData and applications that, with the assistance of AI-related approaches, may collect important information for their businesses. To this aim, however, they must have a solid strategy for collecting, processing, and analyzing data. Thus, the DI process is crucial. It consists of merging information from two or more different sources into a single one. To achieve this goal to its maximum potential, there are still many challenges to be addressed. These are related to the various source data formats, the lack of essential data that can be joined, or the availability of low-quality or not very recent data. Other problems come from having too much data to integrate or from software obsolescence. However, the ability to act independently and automate the integration process is a very important

component. Exactly this factor was taken into account in our initial research project, which was focused on the development of a semi-automated DI process that could operate without the involvement of IT-experts who are often needed to refine the final results of the integration. The developed methodology begins with two heterogeneous sources and determines the tables of the two starting systems be integrated by performing syntactic and semantic analyses on the available data, using an IR approach, a clustering method and a trained neural network. The novelty of our process is that the assistance of IT experts is not necessary, as a specific figure who is not required to have IT skills is involved in the determination of which of the two analyses provides more consistent results for the final integration. The described method has produced a very high accuracy after being tested on different starting systems.

The second topic of this study is related to the ATS algorithms. While these methods are continuously improved as a result of the advancement of AI, finding a metric that can evaluate how well their outputs are, is still a challenging task. Therefore, the goal of our research was to determine whether the ROUGE, which is the metric most frequently used to evaluate the efficacy of these algorithms in the literature, is actually useful for that purpose. For this reason, we conducted a thorough study comparing the performance of many algorithms of both the EATS and AATS types, on a wide range of initial datasets, to verify if this metric is able to distinguish the goodness of one algorithm respect to another. Given that all algorithms provided relatively similar results, the findings demonstrated that this metric is not very inefficient. A similar result was obtained when the initial dataset had only data related to a single topic. In order to complete this work, we also compared the outcomes of some other metrics, also if were used less frequently than the ROUGE for the purpose, which however showed that this challenge is still open.

5.2 FUTURE DIRECTIONS

The two topics that were addressed in this PhD program were the DI and the ATS algorithms evaluation.

As regard the first, the main objective of the future perspective aimed at enhancing the specified method is the conversion of the process from semi-automatic (therefore with the partial involvement of the company manager) to fully automatic. To do this, we intend to develop algorithms capable of autonomously determining which error thresholds and system constraints will be acceptable. In this way, the intermediary role of the company manager as an intermediary will be rendered redundant, and the procedure can be used without the assistance of IT-experts. Furthermore, even the final schemes generated by the integration can be further optimized, by eliminating unnecessary columns, with new smart strategies. As last but not least stage in improving this process, we aim to develop a plan for the one-shot integration of n sources.

For the ATS evaluation research topic, our basic investigation has demonstrated the unsatisfactory of the ROUGE metric for the ATS evaluation purpose, because the way it is built is not sensitive to the context. In fact, until now only human evaluation can accurately determine the goodness of a summary generated by an ATS algorithm. This is because, beyond the limits of the ROUGE metric, the quality of the summaries cannot be distinguished by the semantics also from the other metrics of evaluation, because they capture only the basic relationship between sentences. But even human judgment is not without its flaws. Indeed, if multiple people are asked to rate the quality of a summary, they will all make different choices. This decision may be based on a number of variables relating to the individual conducting the evaluation, including mother tongue, experience, personal judgment and other variables. Therefore, future research will focus on making the assessment as objective as possible, best imitating human judgment and attempting to identify specific features that may relate to readability, coverage of key topics, fluency of language, comprehensibility and correctness of the summary with respect to the source text. We also intend to expand the analysis to incorporate additional, more powerful algorithms. Our ultimate goal is to create a new non-statistical metric to evaluate the quality of summaries, always taking advantage of NLP algorithms for text comprehension.

BIBLIOGRAPHY

[1] Wil MP Van der Aalst. "Data scientist: The engineer of the future." In: *Enterprise interoperability VI*. Springer, 2014, pp. 13–26.

[2] Hervé Abdi. "Singular value decomposition (SVD) and generalized singular value decomposition." In: *Encyclopedia of measurement and statistics* (2007), pp. 907–912.

[3] Mahsa Afsharizadeh, Hossein Ebrahimpour-Komleh, and Ayoub Bagheri. "Query-oriented text summarization using sentence extraction technique." In: *2018 4th international conference on web research (ICWR)*. IEEE. 2018, pp. 128–132.

[4] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[5] Rasim M Alguliyev, Ramiz M Aliguliyev, Nijat R Isazade, Asad Abdi, and Norisma Idris. "COSUM: Text summarization based on clustering and optimization." In: *Expert Systems* 36.1 (2019), e12340.

[6] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. "Text summarization techniques: a brief survey." In: *arXiv preprint arXiv:1707.02268* (2017).

[7] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. "A brief survey of text mining: Classification, clustering and extraction techniques." In: *arXiv preprint arXiv:1707.02919* (2017).

[8] Ioannis Antonellis and Efstratios Gallopoulos. "Exploring term-document matrices from matrix models in text mining." In: *arXiv preprint cs/0602076* (2006).

[9] Abdelkrime Aries, Walid Khaled Hidouci, et al. "Automatic text summarization: What has been done and what has to be done." In: *arXiv preprint arXiv:1904.00688* (2019).

[10] Kirk Baker. "Singular value decomposition tutorial." In: *The Ohio State University* 24 (2005).

[11] Satanjeev Banerjee and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments." In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

[12] Zohra Bellahsene, Angela Bonifati, Fabien Duchateau, and Yannis Velegrakis. "On evaluating schema matching and mapping." In: *Schema matching and mapping*. Springer, 2011, pp. 253–291.

[13] Sonia Bergamaschi, Domenico Beneventano, Alberto Corni, Entela Kazazi, Mirko Orsini, Laura Po, and Serena Sorrentino. "The Open Source release of the MOMIS Data Integration System." In: *SEBD*. Citeseer. 2011, pp. 175–186.

[14] Petr Berka and Ivan Bruha. "Discretization and grouping: Preprocessing steps for data mining." In: *European symposium on principles of data mining and knowledge discovery*. Springer. 1998, pp. 239–245.

[15] Mohammed Binjubeir, Abdulghani Ali Ahmed, Mohd Arfian Bin Ismail, Ali Safaa Sadiq, and Muhammad Khurram Khan. "Comprehensive survey on big data privacy protection." In: *IEEE Access* 8 (2019), pp. 20067–20079.

[16] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[17] Thomas E Booth. "Power iteration method for the several largest eigenvalues and eigenfunctions." In: *Nuclear science and engineering* 154.1 (2006), pp. 48–62.

[18] Faouzi Boufares and A Ben Salem. "Heterogeneous data-integration and data quality: Overview of conflicts." In: *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*. IEEE. 2012, pp. 867–874.

[19] Farid Bourennani, Mouhcine Guennoun, and Ying Zhu. "Clustering relational database entities using k-means." In: *2nd International Conference on Advances in Databases, Knowledge, and Data Applications*. IEEE. 2010, pp. 143–148.

[20] Xiaoyan Cai and Wenjie Li. "A spectral analysis approach to document summarization: clustering and ranking sentences simultaneously." In: *Information Sciences* 181.18 (2011), pp. 3816–3827.

[21] Marie Chavent, Francisco de AT de Carvalho, Yves Lechevallier, and Rosanna Verde. "New clustering methods for interval data." In: *Computational statistics* 21.2 (2006), pp. 211–229.

[22] Laifu Chen and Minh Le Nguyen. "Sentence Selective Neural Extractive Summarization with Reinforcement Learning." In: *11th Intl. Conf. on Knowl. and Sys. Eng. (KSE)*. IEEE. 2019, pp. 1–5.

[23] Yisong Chen and Qing Song. "News text summarization method based on bart-textrank model." In: *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. Vol. 5. IEEE. 2021, pp. 2005–2010.

[24] Jianpeng Cheng, Li Dong, and Mirella Lapata. "Long short-term memory-networks for machine reading." In: *arXiv preprint arXiv:1601.06733* (2016).

[25] Gobinda G Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.

[26] Vipul Dalal and Latesh Malik. "A survey of extractive and abstractive text summarization techniques." In: *6th Inlt. Conf. on Emerging Trends in Eng. and Tech.* IEEE. 2013, pp. 109–110.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805* (2018).

[28] Abhishek Dixit, Vijay Singh Rathore, and Anchal Sehgal. "Improved Google Page Rank Algorithm." In: *Emerging trends in expert applications and security*. Springer, 2019, pp. 535–540.

[29] Hong-Hai Do. "Schema matching and mapping-based data integration." In: (2006).

[30] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of data integration*. Elsevier, 2012.

[31] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. "Automatic text summarization: A comprehensive survey." In: *Expert Systems with Applications* 165 (2021), p. 113679.

[32] Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. "Automatic text summarization: A comprehensive survey." In: *Expert Systems with Applications* 165 (2021), p. 113679. ISSN: 0957-4174.

[33] Günes Erkan and Dragomir R Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." In: *Journal of artificial intelligence research* 22 (2004), pp. 457–479.

[34] Hector Garcia-Molina, Wilburt J Labio, Janet L Wiener, and Yue Zhuge. "Distributed and Parallel Computing Issues in Data Warehousing (Invited Talk)." In: *Proc. of the Tenth Annual ACM Symposium on Parallel Algorithms and Architectures*. Vol. 10. 277651.277673. Citeseer. 1998.

[35] Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. "A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. July 2020, pp. 1302–1308.

[36] Behzad Golshan, Alon Halevy, George Mihaila, and Wang-Chiew Tan. "Data integration: After the teenage years." In: *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI symposium on principles of database systems*. 2017, pp. 101–106.

[37] Derek Greene and Pádraig Cunningham. "Practical solutions to the problem of diagonal dominance in kernel document clustering." In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 377–384.

[38] David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. "A closer look at skip-gram modelling." In: *Proceedings of the fifth international conference on language resources and evaluation (LREC'06)*. 2006.

[39] Laura M Haas, Eileen Tien Lin, and Mary A Roth. "Data integration through database federation." In: *IBM Systems Journal* 41.4 (2002), pp. 578–596.

[40] Alon Halevy, Anand Rajaraman, and Joann Ordille. "Data integration: The teenage years." In: *Proceedings of the 32nd international conference on Very large data bases*. 2006, pp. 9–16.

[41] Xu Han, Tao Lv, Zhirui Hu, Xinyan Wang, and Cong Wang. "Text Summarization Using FrameNet-Based Semantic Graph Model." In: *Sci. Prog.* 2016 (2016).

[42] Puruso Muhammad Hanunggul and Suyanto Suyanto. "The impact of local attention in lstm for abstractive text summarization." In: *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE. 2019, pp. 54–57.

[43] Md Majharul Haque, Suraiya Pervin, and Zerina Begum. "Literature review of automatic single document text summarization using NLP." In: *International Journal of Innovation and Applied Studies* 3.3 (2013), pp. 857–865.

[44] John A Hartigan and Manchek A Wong. "Algorithm AS 136: A k-means clustering algorithm." In: *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[45] Kevin Erich Heinrich, Ramin Homayouni, and Bradford Ryan Silver. *System and method of prediction through the use of latent semantic indexing*. US Patent 10,224,119. 2019.

[46] James Hendler. "Data integration for heterogenous datasets." In: *Big data* 2.4 (2014), pp. 205–215.

[47] Thomas Hofmann. "Probabilistic latent semantic indexing." In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999, pp. 50–57.

[48] Hamidah Ibrahim, Yaser Karasneh, Meghdad Mirabi, Razali Yaakob, and Mohamed Othman. "An Automatic Domain Independent Schema Matching in Integrating Schemas of Heterogeneous Relational Databases." In: *J. Inf. Sci. Eng.* 30.5 (2014), pp. 1505–1536.

[49] Abdul Jabbar, Xi Li, and Bourahla Omar. "A survey on generative adversarial networks: Variants, applications, and training." In: *ACM Computing Surveys (CSUR)* 54.8 (2021), pp. 1–49.

[50] Prabhudas Janjanam and CH Pradeep Reddy. "Text Summarization: An Essential Study." In: *Intl. Conf. on Computational Intelligence in Data Science (ICCIDS)*. IEEE. 2019, pp. 1–6.

[51] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, and Panos Vassiliadis. *Fundamentals of data warehouses*. Springer Science & Business Media, 2002.

[52] Anjali Ganesh Jivani. "A comparative study of stemming algorithms." In: *International Journal of Computer Technology and Applications* 2.6 (2011), pp. 1930–1938.

[53] Anirudh Kadadi, Rajeev Agrawal, Christopher Nyamful, and Rahman Atiq. "Challenges of data integration and interoperability in big data." In: *2014 IEEE international conference on big data (big data)*. IEEE. 2014, pp. 38–40.

[54] Samuel Kaski. "Data exploration using self-organizing maps." In: *Acta polytechnica scandinavica: Mathematics, computing and management in engineering series no. 82*. Citeseer. 1997.

[55] Mohammad Reza Keyvanpour, Mehrnoush Barani Shirzad, and Haniyeh Rashidghalam. "ELTS: A Brief Review for Extractive Learning-Based Text Summarizatoin Algorithms." In: *5th Intl. Conf. on Web Research (ICWR)*. IEEE. 2019, pp. 234–239.

[56] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. "A review of machine learning algorithms for text-documents classification." In: *Journal of advances in information technology* 1.1 (2010), pp. 4–20.

[57] CM Kortman. "Redundancy reduction—a practical method of data compression." In: *Proceedings of the IEEE* 55.3 (1967), pp. 253–263.

[58] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. "Evaluating the factual consistency of abstractive text summarization." In: *arXiv preprint arXiv:1910.12840* (2019).

[59] Stephen Kucer. "The cognitive base of reading and writing." In: *The dynamics of language learning* (1987), pp. 27–51.

[60] Arun Kumar, Matthias Boehm, and Jun Yang. "Data management in machine learning: Challenges, techniques, and systems." In: *Proceedings of the 2017 ACM International Conference on Management of Data*. 2017, pp. 1717–1722.

[61] Krishan Kumar. "Text query based summarized event searching interface system using deep learning over cloud." In: *Multimedia Tools and Applications* 80.7 (2021), pp. 11079–11094.

[62] Swati Kumari and Sanjay Kumar. "A comparative study of various data transformation techniques in data mining." In: *International Journal of Scientific Engineering and Technology* 4.3 (2015), pp. 146–148.

[63] Dawn Lawrie, W Bruce Croft, and Arnold Rosenberg. "Finding topic words for hierarchical summarization." In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. 2001, pp. 349–357.

[64] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents." In: *International conference on machine learning*. PMLR. 2014, pp. 1188–1196.

[65] Jinpeng Li, Chuang Zhang, Xiaojun Chen, Yanan Cao, Pengcheng Liao, and Peng Zhang. "Abstractive text summarization with multi-head attention." In: *2019 international joint conference on neural networks (ijcnn)*. IEEE. 2019, pp. 1–8.

[66] Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. "Deep recurrent generative decoder for abstractive text summarization." In: *arXiv preprint arXiv:1708.00625* (2017).

[67] Hong Liang, Xiao Sun, Yunlei Sun, and Yuan Gao. "Text feature extraction based on deep learning: a review." In: *EURASIP journal on wireless communications and networking* 2017.1 (2017), pp. 1–12.

[68] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm." In: *Pattern Recognition* 36.2 (2003), pp. 451–461.

[69] Chin-Yew Lin and Eduard H. Hovy. "Manual and automatic evaluation of summaries." In: *ACL 2002*. 2002.

[70] Chin-Yew Lin and FJ Och. "Looking for a few good metrics: ROUGE and its evaluation." In: *Ntcir workshop*. 2004.

[71] Hui Lin and Vincent Ng. "Abstractive summarization: A survey of the state of the art." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 9815–9822.

[72] Jerry Chun-Wei Lin, Philippe Fournier-Viger, Lintai Wu, Wensheng Gan, Youcef Djenouri, and Ji Zhang. "PPSF: An open-source privacy-preserving and security mining framework." In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 1459–1463.

[73] Guofeng Liu, Shaobin Huang, and Yuan Cheng. "Research on semantic integration across heterogeneous data sources in grid." In: *Frontiers in Computer Education*. Springer, 2012, pp. 397–404.

[74] Yan Liu, Sheng-hua Zhong, and Wenjie Li. "Query-oriented multi-document summarization via unsupervised deep learning." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 26. 1. 2012, pp. 1699–1705.

[75] Mary ES Loomis. *Data Management and File Structures*. Prentice-Hall, Inc., 1988.

[76] H. P. Luhn. "The Automatic Creation of Literature Abstracts." In: *IBM Journal of Research and Development* 2.2 (1958), pp. 159–165.

[77] Hans Peter Luhn. "The automatic creation of literature abstracts." In: *IBM Journal of research and development* 2.2 (1958), pp. 159–165.

[78] Chuangtao Ma, Bálint Molnár, Adàm Tarcsi, and András Benczùr. "Knowledge Enriched Schema Matching Framework for Heterogeneous Data Integration." In: *2022 IEEE 2nd Conference on Information Technology and Data Science (CITDS)*. 2022, pp. 183–188.

[79] Abhishek Mahajani, Vinay Pandya, Isaac Maria, and Deepak Sharma. "A comprehensive survey on extractive and abstractive techniques for text summarization." In: *Ambient communications and computer systems* (2019), pp. 339–351.

[80] Jonathan I Maletic and Andrian Marcus. "Data Cleansing: Beyond Integrity Analysis." In: *Iq*. Citeseer. 2000, pp. 200–209.

[81] Petros Manousis, Panos Vassiliadis, Apostolos Zarras, and George Papastefanatos. "Schema evolution for databases and data warehouses." In: *European Business Intelligence Summer School*. Springer. 2015, pp. 1–31.

[82] Larry R Medsker and LC Jain. "Recurrent neural networks." In: *Design and Applications* 5 (2001), pp. 64–67.

[83] Osama A Mehdi, Hamidah Ibrahim, and Lilly Suriani Affendey. "An approach for instance based schema matching with google similarity and regular expression." In: *International Arabian Journal Information Technology* 14.5 (2017), pp. 755–763.

[84] Nataliia Melnykova, Uliana Marikutsa, and Uriy Kryvenchuk. "The new approaches of heterogeneous data consolidation." In: *2018 IEEE 13th international scientific and technical conference on computer sciences and information technologies (CSIT)*. Vol. 1. IEEE. 2018, pp. 408–411.

[85] Rada Mihalcea and Paul Tarau. "Textrank: Bringing order into text." In: *Proceedings of the 2004 conference on empirical methods in natural language processing*. 2004, pp. 404–411.

[86] Tomas Mikolov, Kai Chen, Gregory S Corrado, and Jeffrey A Dean. *Computing numeric representations of words in a high-dimensional space*. US Patent 9,037,464. 2015.

[87] N Moratanch and S Chitrakala. "A survey on extractive text summarization." In: *2017 international conference on computer, communication and signal processing (ICCCSP)*. IEEE. 2017, pp. 1–6.

[88] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." In: *arXiv preprint arXiv:1602.06023* (2016).

[89] Emily Namey, Greg Guest, Lucy Thairu, and Laura Johnson. "Data reduction techniques for large qualitative data sets." In: *Handbook for team-based qualitative research* 2.1 (2008), pp. 137–161.

[90] N Nazari and MA Mahdavi. "A survey on automatic text summarization." In: *Journal of AI and Data Mining* 7.1 (2019), pp. 121–135.

[91] Ani Nenkova and Rebecca J Passonneau. "Evaluating content selection in summarization: The pyramid method." In: *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*. 2004, pp. 145–152.

[92] Jun-Ping Ng and Viktoria Abrecht. "Better summarization evaluation with word embeddings for ROUGE." In: *arXiv preprint arXiv:1508.06034* (2015).

[93] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning." In: *Neurocomputing* 452 (2021), pp. 48–62.

[94] Paulo Cesar Fernandes de Oliveira. "How to evaluate the 'goodness' of summaries automatically." PhD thesis. University of Surrey, 2005.

[95] Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, and Ilyas Cicekli. "Text summarization using latent semantic analysis." In: *Journal of Information Science* 37.4 (2011), pp. 405–417.

[96] Aishwarya Padmakumar and Akanksha Saran. "Unsupervised text summarization using sentence embeddings." In: *Technical Report, University of Texas at Austin* (2016), pp. 1–9.

[97] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab, 1999.

[98] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a method for automatic evaluation of machine translation." In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[99] Paolo Papotti, Felix Naumann, Sebastian Kruse, et al. *Systems and methods for data integration*. US Patent 10,528,532. 2020.

[100] Christine Parent and Stefano Spaccapietra. "Issues and approaches of database integration." In: *Communications of the ACM* 41.5es (1998), pp. 166–178.

[101] Darshna Patel, Saurabh Shah, and Hitesh Chhinkaniwala. "Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique." In: *Expert Systems with Applications* 134 (2019), pp. 167–177.

[102] Meetkumar Patel, Adwaita Chokshi, Satyadev Vyas, and Khushbu Maurya. "Machine learning approach for automatic text summarization using neural networks." In: *International Journal of Advanced Research in Computer and Communication Engineering* 7.1 (2018).

[103] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. "Language models as knowledge bases?" In: *arXiv preprint arXiv:1909.01066* (2019).

[104] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. "Selection of K in K-means clustering." In: *Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 219.1 (2005), pp. 103–119.

[105] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners." In: *OpenAI blog* 1.8 (2019), p. 9.

[106] Erhard Rahm and Philip A Bernstein. "A survey of approaches to automatic schema matching." In: *the VLDB Journal* 10.4 (2001), pp. 334–350.

[107] Erhard Rahm and Hong Hai Do. "Data cleaning: Problems and current approaches." In: *IEEE Data Engineering Bulletin* 23.4 (2000), pp. 3–13.

[108] Erhard Rahm and Eric Peukert. *Large-Scale Schema Matching*. 2019.

[109] Chandan K Reddy and Bhanukiran Vinzamuri. "A survey of partitional and hierarchical clustering algorithms." In: *Data clustering*. Chapman and Hall/CRC, 2018, pp. 87–110.

[110] Tohida Rehman, Suchandan Das, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. "Abstractive Text Summarization Using Attentive GRU Based Encoder-Decoder." In: *Applications of Artificial Intelligence and Machine Learning*. Springer, 2022, pp. 687–695.

[111] Banafsheh Rekabdar, Christos Mousas, and Bidyut Gupta. "Generative adversarial network with policy gradient for text summarization." In: *2019 IEEE 13th international conference on semantic computing (ICSC)*. IEEE. 2019, pp. 204–207.

[112] Afsaneh Rezaei, Sina Dami, and Parisa Daneshjoo. "Multi-Document Extractive Text Summarization via Deep Learning Approach." In: *5th Conf. on Knowledge Based Engineering and Innovation (KBEI)*. IEEE. 2019, pp. 680–685.

[113] Fakhitah Ridzuan and Wan Mohd Nazmee Wan Zainon. "A review on data cleansing methods for big data." In: *Procedia Computer Science* 161 (2019), pp. 731–738.

[114] Helge Ritter, Thomas Martinetz, Klaus Schulten, Daniel Barsky, Marcus Tesch, and Ronald Kates. *Neural computation and self-organizing maps: an introduction*. Addison-Wesley Reading, MA, 1992.

[115] Barbara Rosario. "Latent semantic indexing: An overview." In: *Techn. rep. INFOSYS* 240 (2000), pp. 1–16.

[116] Tanvi Sahay, Ankita Mehta, and Shruti Jadon. "Schema matching using machine learning." In: *7th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2020, pp. 359–366.

[117] Kamal Sarkar and Sivaji Bandyopadhyay. "Generating headline summary from a document set." In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2005, pp. 649–652.

[118] Murat H Sazli. "A brief review of feed-forward neural networks." In: *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering* 50.01 (2006).

[119] Natalie Schluter. "The limits of automatic summarisation according to rouge." In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. 2017, pp. 41–45.

[120] Abigail See, Peter J Liu, and Christopher D Manning. "Get to the point: Summarization with pointer-generator networks." In: *arXiv preprint arXiv:1704.04368* (2017).

[121] Doug Semenick. "Tests and measurements: The T-test." In: *Strength & Conditioning Journal* 12.1 (1990), pp. 36–37.

[122] Shail Shah, S Adarshan Naiynar, and B Amutha. "Query based text summarization." In: *ARPN J. Eng. Appl. Sci* 12.19 (2006), pp. 201–206.

[123] Aakanksha Sharaff, Amit Siddharth Khaire, and Dimple Sharma. "Analysing Fuzzy Based Approach for Extractive Text Summarization." In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. 2019, pp. 906–910.

[124] Pavel Shvaiko and Jérôme Euzenat. "A survey of schema-based matching approaches." In: *Journal on data semantics IV*. Springer, 2005, pp. 146–171.

[125] Aisha Siddiqa, Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Mohsen Marjani, Shahabuddin Shamshirband, Abdullah Gani, and Fariza Nasaruddin. "A survey of big data management: Taxonomy and state-of-the-art."

In: *Journal of Network and Computer Applications* 71 (2016), pp. 151–166.

[126] Ajit P Singh and Geoffrey J Gordon. "A unified view of matrix factorization models." In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2008, pp. 358–373.

[127] Amit Singhal et al. "Modern information retrieval: A brief overview." In: *IEEE Data Eng. Bull.* 24.4 (2001), pp. 35–43.

[128] Shengli Song, Haitao Huang, and Tongxiao Ruan. "Abstractive text summarization using LSTM-CNN based deep learning." In: *Multimedia Tools and Applications* 78.1 (2019), pp. 857–875.

[129] Xiaomeng Su et al. "Introduction to big data." In: *IFUD1123* (2017).

[130] Dima Suleiman and Arafat A Awajan. "Deep Learning Based Extractive Text Summarization: Approaches, Datasets and Evaluation Measures." In: *6th Intll. Conf. on Social Networks Analysis, Manag. and Sec. (SNAMS)*. IEEE. 2019, pp. 204–210.

[131] Dima Suleiman and Arafat Awajan. "Deep learning based abstractive text summarization: approaches, datasets, evaluation measures, and challenges." In: *Mathematical problems in engineering* 2020 (2020).

[132] Wencheng Sun, Zhiping Cai, Yangyang Li, Fang Liu, Shengqun Fang, and Guoyan Wang. "Data processing and text mining technologies on electronic medical records: a review." In: *Journal of healthcare engineering* 2018 (2018).

[133] Pamela Szabó. "Data Virtualization and Federation." In: *Stone Bond Technologies* (2014).

[134] Hamed Taherdoost. "Sampling methods in research methodology; how to choose a sampling technique for research." In: *How to choose a sampling technique for research (April 10, 2016)* (2016).

[135] Oguzhan Tas and Farzad Kiyani. "A survey automatic text summarization." In: *PressAcademia Procedia* 5.1 (2007), pp. 205–213.

[136] Ronald A Thisted. "What is a P-value." In: *Departments of Statistics and Health Studies* (1998).

[137] Yves Tillé and Matthieu Wilhelm. "Probability sampling designs: principles for choice of design and balancing." In: *Statistical Science* (2017), pp. 176–189.

[138] Vidyadhar Upadhya and PS Sastry. "An overview of restricted Boltzmann machines." In: *Journal of the Indian Institute of Science* 99.2 (2019), pp. 225–236.

[139] Mladen Varga and Katarina Ćurko. "Some aspects of information systems integration." In: *2012 Proceedings of the 35th International Convention MIPRO*. IEEE. 2012, pp. 1583–1588.

[140] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In: *Adv. in Neural Inf. Processing Systems* 30 (2017), pp. 5998–6008.

[141] Pradeepika Verma and Anshul Verma. "A review on text summarization techniques." In: *Journal of Scientific Research* 64.1 (2020), pp. 251–257.

[142] Sukriti Verma and Vagisha Nidhi. "Extractive summarization using deep learning." In: *arXiv preprint arXiv:1708.04439* (2017).

[143] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. "Pointer networks." In: *Advances in neural information processing systems* 28 (2015).

[144] Li Weng, Gagan Agrawal, Umit Catalyurek, T Kur, Sivaramakrishnan Narayanan, and Joel Saltz. "An approach for automatic data virtualization." In: *Proceedings. 13th IEEE International Symposium on High performance Distributed Computing, 2004.* IEEE. 2004, pp. 24–33.

[145] Colin White. "Data integration: Using etl, eai, and eii tools to create an integrated enterprise." In: *Business Intelligence Journal* 10.I (2005).

[146] Zeyu Xiong, Qiangqiang Shen, Yueshan Xiong, Yijie Wang, and Weizi Li. "New generation model of word vector representation based on CBOW or skip-gram." In: *Computers, Materials and Continua* 60.1 (2019), p. 259.

[147] Jiacheng Xu and Greg Durrett. "Neural extractive text summarization with syntactic compression." In: *arXiv preprint arXiv:1902.00863* (2019).

[148] Lei Xu, Chunxiao Jiang, Jian Wang, Jian Yuan, and Yong Ren. "Information security in big data: privacy and data mining." In: *Ieee Access* 2 (2014), pp. 1149–1176.

[149] Yong Xue, Weitao Liu, Boqin Feng, and Wen Cao. "Merging of Topic Maps based on corpus." In: *International Conference on Electrical and Control Engineering*. IEEE. 2010, pp. 2840–2843.

[150] Yuan Yang, Mengdong Chen, and Bin Gao. "An effective content-based schema matching algorithm." In: *International Seminar on Future Information Technology and Management Engineering*. IEEE. 2008, pp. 7–11.

[151] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. "Hierarchical attention networks for document classification." In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 2016, pp. 1480–1489.

[152] Kaichun Yao, Libo Zhang, Dawei Du, Tiejian Luo, Lili Tao, and Yanjun Wu. "Dual encoding for abstractive text summarization." In: *IEEE transactions on cybernetics* 50.3 (2018), pp. 985–996.

[153] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. "Graph-based neural multi-document summarization." In: *arXiv preprint arXiv:1706.06681* (2017).

[154] KangHui Ying, WenYu Hu, Jin Bo Chen, and Guo Nong Li. "Research on instance-level data cleaning technology." In: *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*. 2021, pp. 238–242.

[155] Jing Zhang, Bonggun Shin, Jinho D Choi, and Joyce C Ho. "SMAT: An attention-based deep learning solution to the automation of schema matching." In: *European Conference on Advances in Databases and Information Systems*. Springer. 2021, pp. 260–274.

[156] Shengnan Zhang, Yan Hu, and Guangrong Bian. "Research on string similarity algorithm based on Levenshtein Distance." In: *2nd International Conference on Advanced Information Technology, Electronic and Automation Control (IAEAC)*. IEEE. 2017, pp. 2247–2251.

[157] Patrick Ziegler and Klaus R Dittrich. "Data integration—problems, approaches, and perspectives." In: *Conceptual modelling in information systems engineering*. Springer, 2007, pp. 39–58.