AUTONOMOUS UNIVERSITY OF THE STATE OF MEXICO

TIANGUISTENCO PROFESSIONAL ACADEMIC UNIT

EVALUATION OF TEXT SUMMARIES WITHOUT HUMAN REFERENCES BASED ON DOCUMENT COMPLEXITY ANALYSIS

THESIS

TO OBTAIN THE DEGREE OF
PhD. IN COMPUTER SCIENCE

BY:
M.C.S. JONATHAN ROJAS SIMÓN

ADVISER:
PhD. YULIA NIKOLAEVNA LEDENEVA

CO-ADVISORS:
PhD. RENÉ ARNULFO GARCÍA HERNÁNDEZ
PhD. ÁNGEL HERNÁNDEZ CASTAÑEDA

TIANGUISTENCO, STATE OF MEXICO MAY 2023

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

UNIDAD ACADÉMICA PROFESIONAL TIANGUISTENCO

EVALUACIÓN DE RESÚMENES DE TEXTO
SIN REFERENCIAS HUMANAS BASADA EN EL
ANÁLISIS DE COMPLEJIDAD DE DOCUMENTOS

TESIS

PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

QUE PRESENTA:
M.C.C. JONATHAN ROJAS SIMÓN

TUTORA ACADÉMICA:
DRA. YULIA NIKOLAEVNA LEDENEVA

TUTORES ADJUNTOS:
DR. RENÉ ARNULFO GARCÍA HERNÁNDEZ
DR. ÁNGEL HERNÁNDEZ CASTAÑEDA

TIANGUISTENCO, ESTADO DE MÉXICO MAYO 2023

Abstract

Automatic Text Summarization (ATS) plays an essential role in the management of textual information since it condenses the volume of text documents to produce summaries. Nowadays, the development of ATS is constantly growing because we can measure the performance of proposed methods through the Evaluation of Text Summaries (ETS), but evaluating summaries is a complex process. In the state-of-the-art, the ETS has traditionally been performed through the ROUGE system to analyze summaries' content automatically. However, without human-made summaries (human references), the evaluation cannot be carried out. For this reason, the evaluation of summaries without human references has been proposed.

Over the last two decades, the scientific community has proposed methods that do not depend on human references by using the source text as a reference document. In this sense, ROUGE-C, LSA, and SIMetrix have been widely used methods that fulfill this feature. However, they tend not to correlate highly with human assessment. Therefore, optimizing their individual measures via linear optimization has been proposed in previous works (*e.g.*, SECO-SEVA), providing a closer evaluation of human judgments. Although such optimization enabled improvements in automatic evaluation, it involved the adjustment of the parameters of each measure, assuming the presence of different complexity levels in text documents and assessment measures. Thus, the performance of each method varies according to the complexity level of each source document.

In document analysis and information retrieval, text complexity has been addressed from multiple perspectives, such as readability, vocabulary, and the quantity of information they provide (informativeness). In general, text documents are characterized by varying the before-mentioned features because they come from different sources of information. Therefore, any process of generation or evaluation of texts can vary. As a result, text complexity indexes have not been used in the ETS without human references to select the most appropriate measure to evaluate each summary (candidate summary).

This thesis proposes using a methodology of six steps to select appropriate evaluation measures according to the source documents' complexity level. The proposed selection combines 31 measures derived from ROUGE-C, LSA, and SIMetrix methods, which use state-of-the-art techniques focused on content analysis. Across different experimentations done with a Genetic Algorithm (GA) and Multilayer Perceptron (MLP), the results of the proposed selection show correlation improvements concerning other evaluation methods on well-standardized datasets, such as DUC01 and DUC02.

Resumen

La Generación Automática de Resúmenes (GAR) juega un papel esencial en la gestión de la información textual, ya que se condensa el volumen de los documentos de texto para generar resúmenes. Hoy en día, el desarrollo de la GAR está en constante crecimiento porque se puede medir el desempeño de los métodos propuestos mediante la Evaluación de Resúmenes de Texto (ERT), pero evaluar resúmenes es un proceso complejo. En el estado del arte, la ERT se ha realizado tradicionalmente a través del sistema ROUGE para analizar el contenido de los resúmenes de manera automática. Sin embargo, sin resúmenes escritos por humanos (referencias humanas), no se puede llevar a cabo la evaluación. Por esta razón, se ha propuesto la evaluación de resúmenes sin referencias humanas.

A través de las últimas dos décadas, la comunidad científica ha propuesto métodos que no dependen de referencias humanas al utilizar el texto fuente como documento de referencia. En este sentido, ROUGE-C, LSA y SIMetrix han sido métodos ampliamente utilizados que cumplen esta característica. Sin embargo, tienden a no correlacionarse altamente con la evaluación manual. Por lo tanto, en trabajos previos se ha propuesto optimizar sus medidas individuales a través de la optimización lineal (*p. ej.*, SECO-SEVA), proporcionando una evaluación más cercana a los juicios humanos. Si bien dicha optimización permitió mejoras en la evaluación automática, implicó el ajuste de parámetros de cada medida, asumiendo la presencia de diferentes niveles de complejidad en los documentos de texto y medidas de evaluación. Por lo tanto, el desempeño de cada método varía de acuerdo con el nivel de complejidad de cada documento fuente.

En el análisis de documentos y la recuperación de información, la complejidad del texto se ha abordado desde múltiples perspectivas, como la legibilidad, el vocabulario y la cantidad de información que aportan (informatividad). En general, los documentos de texto se caracterizan por tener variaciones en las características antes mencionadas debido a que provienen de diferentes fuentes de información. Por lo tanto, cualquier proceso de generación o evaluación de textos puede variar. En consecuencia, los índices de complejidad de texto no han sido

utilizados en el ETS sin referencias humanas para seleccionar medidas de evaluación adecuadas para evaluar cada resumen (resumen candidato).

En esta tesis se propone una metodología de seis pasos que busca seleccionar las medidas de evaluación apropiadas en función del nivel de complejidad de los documentos fuente. La selección propuesta combina 31 medidas derivadas de los métodos ROUGE-C, LSA y SIMetrix, que utilizan técnicas del estado del arte enfocadas en el análisis de contenido. A través de diferentes experimentaciones realizadas con un Algoritmo Genético (AG) y Perceptrón Multicapa (MLP), los resultados de la selección propuesta muestran mejoras en la correlación con respecto a otros métodos de evaluación en los conjuntos de datos mejor estandarizados, tales como DUC01 y DUC02.

Contents

List of Figures

List of Tables

List of Equations

CHAPTER 1.

Introduction

In recent times, digital text documents have been an essential resource for users in many aspects of their lives. Although the rapid availability of these documents on the Internet has brought advantages in terms of accessibility, its proliferation has caused an overload that requires state-of-the-art alternatives that allow the understanding of much information as possible (Ermakova et al., 2019; Neri-Mendoza et al., 2020). Because of this situation, Automatic Text Summarization (ATS) is a suitable solution that seeks to reduce the volume of text documents through computer programs, presenting the most relevant content for users (see Figure 1) (El-Kassas et al., 2021; Hovy, 2015; Torres-Moreno, 2014).

The ATS is one of the most studied research tasks in Natural Language Processing (NLP), starting with Luhn's (Luhn, 1958) and Edmundson's (Edmundson, 1969) contributions in the '50s and '60s. Nevertheless, the most significant efforts have been carried out over the last two decades, whose proposed techniques and approaches aim to create human-like summaries (El-Kassas et al., 2021; Gambhir & Gupta, 2017; Saggion & Poibeau, 2013). In each of these studies, the Evaluation of Text Summaries (ETS) is considered a fundamental stage that determines the performance of ATS methods (Rojas-Simon et al., 2022). However, the ETS is complex because it faces a huge variety of natural language expressions to express the same ideas (ambiguity) (Gelbukh & Calvo, 2018; Lloret et al., 2018).

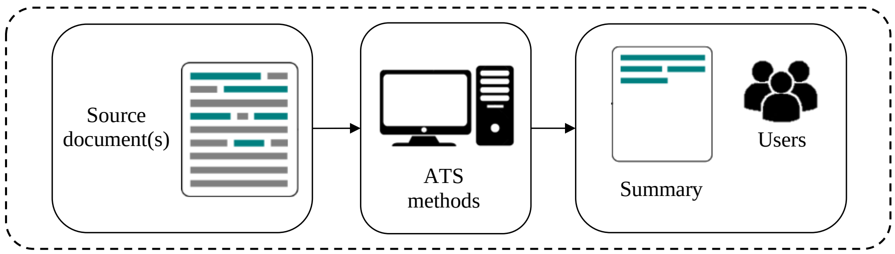


Figure 1. Automatic Text Summarization (Rojas-Simon et al., 2022).

In literature, the ETS is seen as an immersed process of ATS that employs different criteria to assess summaries. According to Jones & Galliers (1995), the ETS has evaluated summaries under *extrinsic* and *intrinsic* criteria.

The *extrinsic evaluation* measures the usefulness of a summary in the context of other tasks, such as Document Categorization (DC), Information Retrieval (IR), and Question Answering (QA) (Lloret et al., 2018; Steinberger & Ježek, 2009). The state-of-the-art *TIPSTER Text Summarization Evaluation* (SUMMAC) workshop has promoted the development of extrinsic evaluation methods (Mani et al., 1999, 2002).

On the other hand, *intrinsic evaluation* analyzes the content and coherence of a summary (Lloret et al., 2018). Generally, intrinsic evaluation tends to compare the textual content of an automatic/manual summary to be evaluated (candidate summary) with one or several reference summaries created by the human expert in the domain of texts and in making summaries (human references) (Louis & Nenkova, 2013; Torres-Moreno et al., 2010), as shown in Figure 2. For several years, intrinsic evaluation has been supported by DUC[1] and TAC[2] workshops (Dang & Owczarzak,

[1] Document Understanding Conferences
[2] Text Analysis Conference

2008; Over et al., 2007). Both workshops have been organized by the National Institute of Standards and Technology (NIST) to promote the ATS and ETS development.

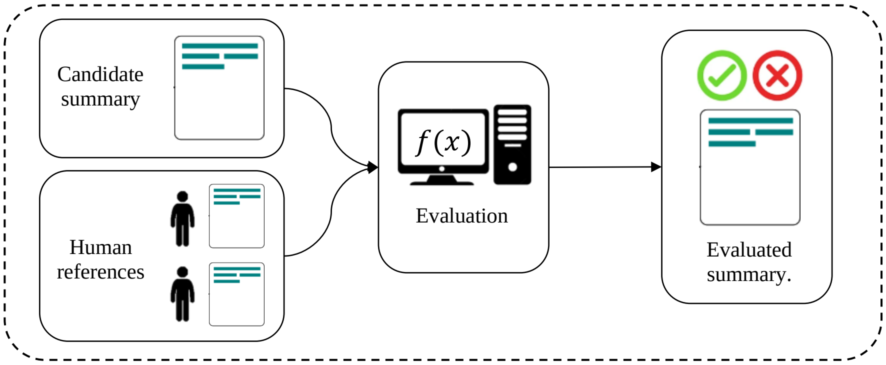


Figure 2. Traditional evaluation (candidate summary - human references) (Rojas-Simon et al., 2022).

According to the degree of human intervention, intrinsic and extrinsic evaluation methods are classified into *manual*, semi-*automatic*, and *automatic*.

Initially, summaries were manually evaluated, requiring human judgments to qualify summaries. The Summary Evaluation Environment (SEE) is the first manual evaluation method to compare the candidate summary content against a set of human references (Lin & Hovy, 2002; Lin, 2001; Over et al., 2007). However, the use of SEE implied a high cost for large-scale workshops because it requires that several human experts (assessors) evaluate for prolonged times.

Afterward, semi-automatic evaluation was proposed, whose methods seek to reduce the cost and difficulty of manual evaluation. Particularly, *Pyramid* is a well-known method that elaborates a selection of SCUs[3] from a set of human references to construct a hierarchical representation of contents through a pyramid of contents (Nenkova & Passonneau, 2004). This representation is used to weigh the most relevant sentences from candidate summaries (Nenkova et al., 2007).

At the same time, the automatic evaluation has been proposed through the implementation of similarity measures (Jing et al., 1998; Radev et al., 2003; Saggion et al., 2002), the extraction of different text representation models (*e.g.*, n-grams) (Lin & Hovy, 2003); the implementation of NLP packages, such as the BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002), Basic Elements (Hovy et al., 2005, 2006), AUTOmatic SUMMary Evaluation based on N-gram

[3] Summary Content Units

Graphs (AutoSummENG) (Giannakopoulos et al., 2008; Giannakopoulos & Karkaletsis, 2011), among others. Nowadays, the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is the most used evaluation package by scientific community, due to its high correlation of their methods[4] with the manual assessment (Lin, 2004). Basically, the ROUGE's methods measure the overlap of terms between a candidate summary and human references. However, these methods do not evaluate summaries if their human references are not available (Cabrera-Diego & Torres-Moreno, 2018).

As a result of this problem, the ETS without human references has been proposed. Particularly, this kind of evaluation has gained interest because it seeks to compare content between the candidate summary and its source document(s) (Rojas-Simon et al., 2022), as shown in Figure 3. In recent years, state-of-the-art methods have been proposed, such as ROUGE-C (He et al., 2008), Latent Semantic Analysis (LSA) (Steinberger & Ježek, 2009), and SIMetrix (Summary Input similarity Metrics), but they do not yet highly correlate to human judgments (Cabrera-Diego et al., 2016).

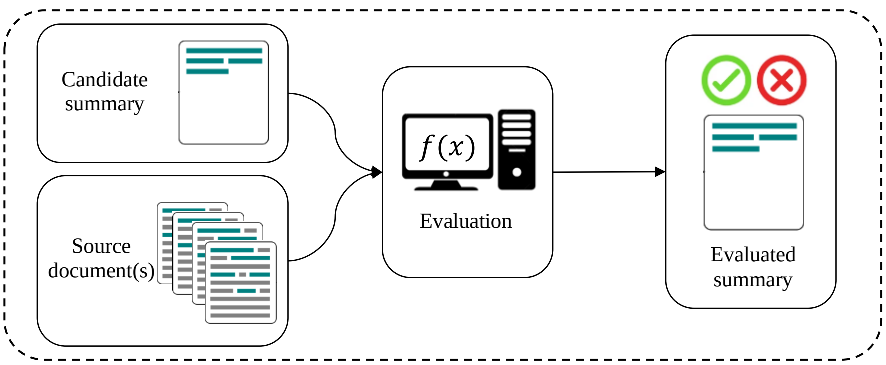


Figure 3. Evaluation of summaries without human references (Rojas-Simon et al., 2022).

In addition to the methods mentioned above, the ETS without human references has focused on combining and optimizing different evaluation measures. Regarding this approach, the representativeness of documents and evaluation techniques are essential to generate several evaluation measures. On the one hand, the representativeness of documents allows for retrieving linguistic and statistical features from the candidate summary and its source document. On the other hand,

[4] ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S are methods of the ROUGE system.

evaluation techniques employ different criteria that calculate similarities among both documents. Based on this idea, a linear combination of measures has been proposed as an alternative to improve the proximity between automatic evaluation and manual assessment.

Early optimization-based evaluation studies have been centered on the combination of ROUGE metrics. For instance, Conroy & Dang (2008) proposed ROUGE Optimal Summarization Evaluation (ROSE) as an evaluation method based on the combination of seven well-established metrics from ROUGE (ROUGE-1, 2, 3, 4, L, SU4, and BE) via canonical correlation analysis (Härdle & Simar, 2015). On the other hand, Ellouze et al. (2016) proposed a combination of 21 linguistic and content-based metrics through linear regression. In general, both approximations have been proposed under the assumption that different metrics may capture several qualities on which an assessor evaluates a summary.

Another way to combine evaluation metrics has been through linear optimization. In previous works (Rojas-Simon et al., 2022; Rojas-Simón et al., 2021), a linear optimization of 31 automatic evaluation measures was performed via Genetic Algorithm (GA), of which seven of them come from ROUGE-C, 16 from LSA, and eight from SIMetrix. The result of this optimization is called SECO-SEVA (SElf-COntent Summary EVAluation) (Rojas-Simón et al., 2023). Unlike (Conroy & Dang, 2008) and (Ellouze et al., 2016), it was proposed that the evolutionary optimization of the GA can provide a combination of measures with a better approximation to the manual assessment. However, the GA's evolutionary adjustment assumes the presence of different levels of complexity in each measure.

Over DUC and TAC workshops, several collections of documents have been provided by different information sources (Dang, 2010; Dang & Owczarzak, 2008, 2009; Over et al., 2007; Owczarzak & Dang, 2011). For instance, some documents come from news agencies such as the Financial Times[5], Associated Press[6], and the Wall Street Journal[7]. Generally, each document shows different levels of complexity due to its redaction and style. Therefore, manual assessment

[5] Financial Times website: https://www.ft.com/
[6] Associated Press website: https://apnews.com/
[7] Wall Street Journal website: https://www.wsj.com/

can behave in different ways under ease/difficulty of readability and style of the source document(s).

In consequence, it is clear to observe that automatic evaluation requires internal parameters adjustment (*e.g.*, text representation models). Such adjustment of parameters seeks that any proposed method/approach is able to maintain a near evaluation to human judgments across all candidate summaries. Nevertheless, the performance of evaluation methods varies according to the complexity level of each source document.

The complexity of text documents has been addressed via readability indexes, which calculate how eassily a document can be read and understood (Zamanian & Heydari, 2012). Currently, an initiative of U.S. denominated *The Common Core State Standards* (CCSS) has used readability indexes to determine the complexity level of texts, thereby helping students to enhance their reading skills (Williamson et al., 2013). In the state-of-the-art, a wide variety of readability indexes have been proposed; however, the Automatic Readability Index (ARI) and Coleman-Liau Index (CLI) are distinguished by using independent-language features, such as the average number of characters by word and the average number of words by sentence.

On the other hand, readability is not the only feature that determines the level of complexity in text documents. In information retrieval and complexity of language, other indexes related to the quantity of information and the morphology of the language have been used. Some of them are shown below.

- Type-Token Relationship (TTR) (Gutierrez-Vasques & Mijangos, 2019)
- Ratio of Stop-Words (RSW) (Yuang et al., 2012)
- Ratio of Inflected Words (RIW) (Vania & Lopez, 2017)
- Average of Characters per Word (ACW) (Ellouze et al., 2017a)
- Average of Words per Sentence (AWS) (Ellouze et al., 2017a)
- Relation of sentences with title (RTF_s) (Vázquez et al., 2018)
- Entropy of words (H_w) (Bentz et al., 2016)
- Entropy of sentences (H_s) (Rankel et al., 2012)

Concerning the before-mentioned text-document complexity indexes, it is necessary to consider that in the evaluation stage, there are candidate summaries generated from source documents with high/low complexity. Therefore, text summaries must be evaluated by a selection of measures that adjust according to the complexity level of each source document.

For the handwritten Text Line Segmentation (TLS), García-Calderón et al. (2019) proposed a complexity index to determine the most appropriate TLS method for each handwritten document. In this work, TLS methods were selected based on several features of handwritten documents, such as the amount of horizontal/vertical information in interlinear spaces, the contrast of colors in documents, and the color of the ink. However, text-document complexity indexes have not been used to enhance ETS's performance without human references.

1.1 Problem statement

The ETS is a fundamental task of ATS that allows calculating the quality of text summaries through manual, semi-automatic, and automatic methods. Regarding automatic evaluation, the scientific community has used the ROUGE system due to the high correlation of their methods concerning manual assessment. Nevertheless, ROUGE's methods do not evaluate summaries without human references (Cabrera-Diego & Torres-Moreno, 2018).

Currently, the ETS without human references has been an alternative to avoid the difficulty and cost that imply generating human references. In the state-of-the-art, methods that fulfill this characteristic have been proposed (*e.g.*, ROUGE-C, LSA, and SIMetrix) based on analyzing information from source documents. Moreover, their combination and optimization have been proposed to improve automatic evaluation (Rojas-Simón et al., 2021), creating SECO-SEVA as an evaluation package used for research purposes. However, the adjustment of each measure supposes the presence of different levels of complexity in source documents.

The complexity of text documents has been addressed under two experimental approximations. The first one depends on the use of readability indexes, which help to determine the degree of readable texts. In contrast, the second approximation uses internal and shallow text features. Nonetheless, such complexity indexes have not been used in the ETS without human references to select the most appropriate measure according to the complexity of source documents.

Regarding the statements previously mentioned, the following research question is presented:
What is the effect of text complexity indexes in the selection of evaluation metrics for the ETS without human references?

1.2 Objectives

1.2.1 General objective

Based on the state-of-the-art and the problem statement, the following general objective is presented:

Determine appropriate evaluation metrics through the level of complexity of source documents to improve the performance of the ETS without human references.

1.2.2 Specific objectives

In order to determine the most appropriate evaluation measures of each candidate summary, the following specific objectives are considered:

- Analyze and pre-process candidate summaries, source documents, and manual evaluation results from DUC01 and DUC02 collections.
- Calculate the level of complexity of each source document from DUC01 and DUC02 collections using the before-mentioned indexes of readability, productivity of language, and quantity of information.
- Evaluate candidate summaries of DUC01 and DUC02 collections with the evaluation metrics used in (Rojas-Simón et al., 2021).
- Determine ranges of complexity of each metric used in (Rojas-Simón et al., 2021) according to the incorporation of readability, productivity of language, and quantity of information indexes.
- Compare the relationship between the variability of evaluation of each metric used in (Rojas-Simón et al., 2021) and the level of complexity in source documents through the Pearson, Spearman, and Kendall correlation coefficients.
- Select the most appropriate metric for each candidate summary using the complexity level of the source document.
- Calculate the correlation between the selection of metrics and the results of manual evaluation via Pearson, Spearman, and Kendall correlations.

1.3 Hypothesis

The hypothesis of this thesis is shown below:

If appropriate evaluation metrics are selected based on the level of complexity from source documents, the performance of the ETS without human references will be improved.

1.4 Organization of the document

This thesis is organized into six chapters. Chapter 1 introduces a general overview of ETS; some definitions of manual, semi-automatic, and automatic evaluation are presented. Moreover, the research problem, the objectives, and the hypothesis of this thesis are presented. Chapter 2 describes the theoretical framework of ETS to understand the main concepts of this task and the basic ideas of automatic evaluation methods. Next, this chapter presents and describes text complexity indexes used in this work. Finally, it includes some concepts related to GA and MLP. Chapter 3 summarizes the related automatic evaluation methods. Chapter 4 presents the proposed methodology to perform a selection of evaluation methods. Chapter 5 describes the used datasets to evaluate summaries. Moreover, the experimental configuration of ROUGE-C, LSA, and SIMetrix is presented; the GA configuration and the correlation results of the optimized automatic evaluation metric are presented. Finally, Chapter 6 presents the conclusions, publications derived from this thesis, and future works.

CHAPTER 2.

Theoretical Framework

Currently, the management of textual information on the Internet has become a challenge for the industry and research over the last two decades. Due to its rapid proliferation, the need to obtain nuggets of information promotes the development of computerized methods that generate summaries automatically. In this regard, Automatic Text Summarization (ATS) has been a significant research mainstream in Natural Language Processing (NLP). Nevertheless, the growth and scope of the ATS are due to the implementation of evaluation methods able to qualify the summaries' quality. This chapter describes the background behind the Evaluation of Text Summaries (ETS) in-depth. The main intention of this chapter is to introduce the reader to the basic ideas and historical context of the ETS across time.

2.1 The Role of Evaluation for NLP and NLG

According to Lloret et al. (2018), evaluation is an essential step for any research task because it is necessary to measure and compare the performance of investigated approaches. In NLP, evaluation is highly required because it helps to determine how suitable a method, approach, or application can be. For example, we can measure the predictability of Sentiment Classification (SC) systems concerning human judgments (also known as *gold standards*) via Information Retrieval (IR) measures, such as Recall, Precision, Accuracy, and F-measure (Liu, 2015; Manning et al., 2009). Figure 4 illustrates the evaluation of an SC system.

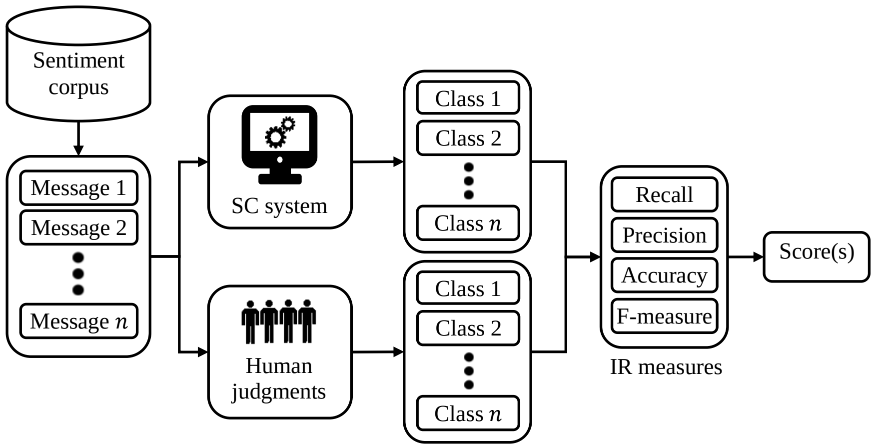


Figure 4. Evaluation of SC systems using IR measures.

However, evaluation is even more difficult in Natural Language Generation (NLG)[8] tasks because there are no correct and unique outputs (Jones & Galliers, 1995; Lloret et al., 2018). On the contrary, there are many valid and possible outputs because the natural language allows variations of expressions, meanings, and ideas in general (Torres-Moreno, 2014). Therefore, these variations require a deep analysis that "understands" the natural language. As shown in Figure 5, the NLG system receives text information from a corpus/dataset, generating m outputs in natural language. Afterward, the evaluation of each output is done by comparing it to a set of n references. Finally, the result of each evaluation is presented in an overall score that reflects the degree of similarity between each output to its corresponding reference.

[8] NLG is the process of automatically generating human-understandable text in one or more natural languages (Sai et al., 2023). In this sense, the Machine Translation (MT), ATS, Question Answering (QA), and Dialogue Generation (DG) are well-known examples of NLG tasks.

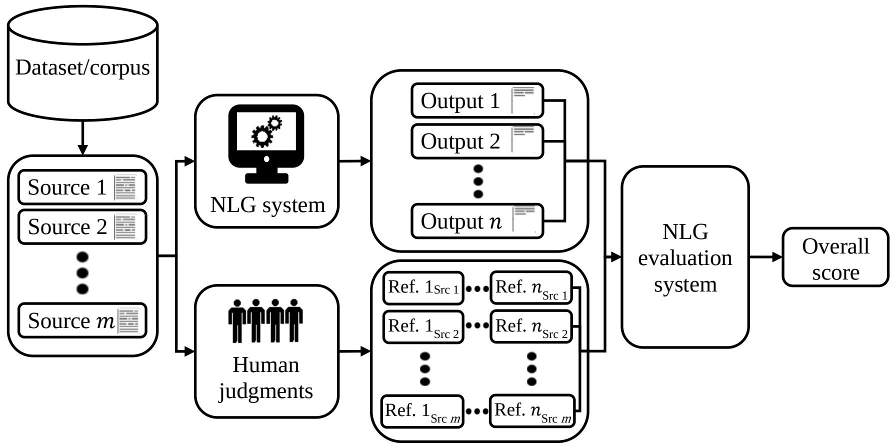


Figure 5. Evaluation of NLG systems.

2.2 Evaluation of Text Summaries (ETS)

In the context of Text Summarization, the evaluation is a fundamental stage to determine the performance of the ATS methods (Lloret et al., 2018). The Evaluation of Text Summaries (ETS) is a process within the ATS that determines the quality of summaries, using pre-established criteria based on specialized algorithms, similarity/distance measures, or techniques (Fiori, 2014), in order to create a score that performs similar to human judgments (Cabrera-Diego et al., 2016; Hovy et al., 2006). These criteria respond to a set of features where a summary must fulfill, which in turn may vary according to the requirements of users (Jones & Galliers, 1995; Lloret et al., 2018).

2.3 Types of Evaluation

As mentioned above, the ETS employs several pre-established criteria on which a summary is evaluated according to linguistic, content, or prospection features. In general, the ETS considers the evaluation of summaries under two main criteria: extrinsic and intrinsic (Jones & Galliers, 1995).

2.3.1 Extrinsic Evaluation

One of the most common evaluation tasks performed by humans is extrinsic evaluation. In the extrinsic evaluation, the summary is evaluated according to its usefulness in another task or context (Lloret et al., 2018; Steinberger & Ježek, 2009). This evaluation is strongly linked to how a summary must fulfill the purpose of other tasks, because of people tends to pragmatically assess a summary (Torres-Moreno, 2014).

The most important and typical tasks of the extrinsic evaluation are Document Categorization (DC), IR, and QA (Steinberger & Ježek, 2009), but other tasks have been recently included, such as Reading Comprehension (RC), Relevance Assessment (RA) and Web Search (WS) (Cabrera-Diego et al., 2016; Lloret et al., 2018).

In the state-of-the-art, extrinsic evaluation methods have been developed on large-scale evaluation workshops, such as the TIPSTER Text Summarization Evaluation Conference (SUMMAC)[9] (Lloret et al., 2018; Mani et al., 1999, 2002). Moreover, other evaluation programs like NTCIR have organized extrinsic evaluation tasks in the third NTCIR workshop (NTCIR-3). The NTCIR-3 was focused on evaluating information extraction systems (IR task), Question-Answering systems (QA task), and Automatic Text Summarization systems (ATS task).

2.3.2 Intrinsic Evaluation

Another evaluation criterion that has been widely researched by the scientific community is the intrinsic evaluation of summaries (Rojas-Simon, 2017). The intrinsic evaluation is focused on the evaluation of the summarization approach in itself, analyzing the *content*, *coherence* and *informativeness* of the summary (Lloret et al., 2018; Mani et al., 2002). According to (Steinberger & Ježek, 2009), intrinsic evaluation is divided into *text quality* and *content* evaluation. The evaluation of text quality involves the assessment of coherence in terms of readability, lapses in grammatically, presence of dangling anaphors, among others (Mani et al., 2002). On the other hand, content-based evaluation measures the ability to identify the essential information of a document from key topics, concepts, or definitions (Steinberger & Ježek, 2009).

Generally, the intrinsic evaluation relates the textual content of a summary to be evaluated (candidate summary) with one (or several) reference summaries made by the human expert (human references) (Lloret et al., 2018; Torres-Moreno et al., 2010). However, due to human references are not always available, other approaches have been proposed that evaluate the content of the

[9] SUMMAC website: https://www-nlpir.nist.gov/related_projects/tipster_summac.

candidate summary concerning its source document(s) (Cabrera-Diego et al., 2016; Cabrera-Diego & Torres-Moreno, 2018).

In the context of intrinsic evaluation, several methods based on content analysis have been proposed on DUC and TAC workshops organized by the National Institute of Standards and Technology (NIST)[10] (Dang & Owczarzak, 2008; Over et al., 2007). Through the DUC's and TAC's workshops, intrinsic content-based evaluation metrics have been proposed, such as Coverage, Weighted Retention, and Pyramids (Lin & Hovy, 2002; Over et al., 2007), which we will describe in-depth in Chapter 3. On the other hand, summaries have also been evaluated through linguistic features, such as grammaticality, cohesion, and organization (Over et al., 2007).

2.4 Manual evaluation

Initially, the evaluation of summaries was manually performed. During the first DUC evaluation workshops, the manual evaluation was carried out by NIST assessors to determine the quality of summaries (Lin & Hovy, 2002). However, it is expensive because of three issues presented in (Amigó, 2006; da Cunha, 2008; Lloret et al., 2018), which are the following:

1. Manual evaluation is time-consuming.
2. Several assessors are required to evaluate a large collection of summaries.
3. Each assessor can evaluate summaries from different perspectives or criteria.

The first manual evaluation task was carried out on DUC-2001 through the Summary Evaluation Environment (SEE). Basically, SEE is an interface used to manually compare the candidate summary (also known as *Peer Summary*) content against a set of human references (also known as *Model Summary*) (see Figure 6) (Lin & Hovy, 2002; Over et al., 2007).

In SEE, each document is decomposed and analyzed into a list of *meaning units* (*e.g.,* sentences or paragraphs) that reflect key ideas or concepts. Such units are highlighted in different colors to provide the user a better overview of both documents. The meaning units marked from the candidate summary are named Peer Units (PUs), and the units marked from the human references are named Model Units (MUs) (Over et al., 2007). Afterward, all PUs is compared to all MUs for

[10] NIST website: https://www.nist.gov.

specifying how similar they are in a list of five descriptions: *all*, *most*, *some*, *hardly any*, and *none*. In summary, this feature presents advantages when there are not the same sentence matches between documents. In addition to the analysis of coincidences between documents, annotators can read candidate summaries to provide overall judgments in terms of grammaticality, cohesion, and organization.

SEE has been used in the first four large-scale evaluation workshops of DUC (DUC-2001 to DUC-2004), generating manual evaluation metrics (Over et al., 2007). Some of them are used in the state-of-the-art as reference manual evaluations to compare the performance of automatic evaluation metrics, with the purpose of obtaining an automatic evaluation metric able to capture several aspects of manual evaluation (Lloret et al., 2018). The most common manual evaluation metrics are described below.

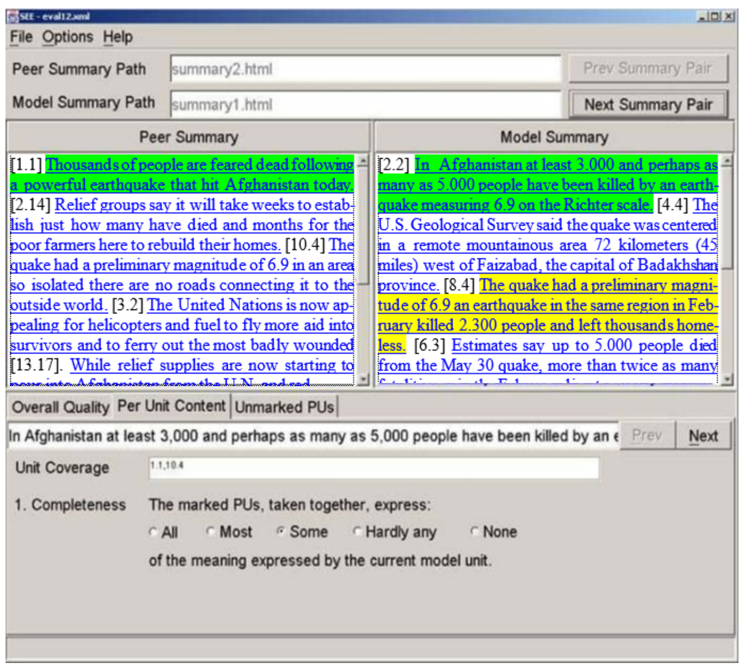


Figure 6. SEE interface (Lin & Hovy, 2002; Over et al., 2007).

2.4.1 Linguistic-based Metrics

Manual evaluation is not only limited to a quantitative analysis of candidate summaries' quality. In view of this, an annotator can perform a qualitative description of summaries using linguistic criteria. Since 2001, three linguistic criteria (grammaticality, cohesion, and organization) were considered for providing a detailed diagnosis of candidate summaries (Over et al., 2007), but in 2005 they were redesigned and expanded to five. Such criteria are defined by NIST as follows:

1 Grammaticality: The summary should have no datelines, system-internal formatting, capitalization errors, or ungrammatical sentences (*e.g.*, fragments and missing components) that make the text difficult to read.
2 Non-redundancy: There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, repeated facts, or the repeated use of a noun phrase (*e.g.*, “Bill Clinton” when a pronoun (“he”) would suffice).
3 Referential clarity: It should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced, but its identity or relation to the story remains unclear.
4 Focus: The summary should have a focus; sentences should only contain information that is related to the rest of the summary.
5 Structure and coherence: The summary should be well-structured and well-organized. The summary should not just be a heap of related information but should build from sentence to sentence to a coherent body of information about a topic.

Each criterion considers a set of five descriptions to indicate the degree of accomplishment (qualification) of each summary: 1. Very poor, 2. Poor, 3. Barely acceptable, 4. Good, 5. Very good. The highest qualification (5) indicates that the summary is produced by a human.

In automatic evaluation, the implementation of linguistic metrics presents a series of issues because the variety of meanings in the text is difficult to understand by computers. Therefore, the before-mentioned criteria are considered only for manual evaluation.

2.4.2 Content-based Metrics

The variability of expressions and meanings of natural language has been an issue that is continuously studied by the state-of-the-art. In the ETS, this situation causes summaries to be difficult to evaluate because the same information can be written differently. The following sentences express the same meaning, but they are written in different ways:

1 Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.
2 Tropical Storm Gilbert was shaped within the eastern Caribbean and fortified into a typhoon Saturday night.
3 Tropical Storm Gilbert developed in the eastern Caribbean and became a hurricane on Saturday night.
4 Tropical Storm Gilbert in the eastern Caribbean Sea strengthened into a hurricane Saturday night.

As a result, content evaluation is a challenging task that seeks to measure the degree of *informativeness* or *content coverage* of summaries even if they present the same information in different ways. Basically, this type of evaluation is performed by comparing content between the summary and human references. To perform an analysis of content, weighted retention and coverage are metrics commonly used in the manual evaluation.

2.4.2.1 Weighted Retention

Weighted retention ($Retention_w$) is defined by means of the relation between the number of Model Units (MUs[11]) found in the candidate summary, and the number of MUs from human references (Lin & Hovy, 2002). This relation is defined in Equation 1.

$$Retention_w = \frac{(Number\ of\ MUs\ found\ in\ the\ candidate\ summary) \times C}{Total\ number\ of\ MUs\ in\ human\ references}$$

Equation 1. $\boldsymbol{Retention_w}$ metric (Lin & Hovy, 2002).

$Retention_w$ is a recall-based metric. The variable C represents the ratio of coverage of the candidate summary; C: 1 for *all*, 3/4 for *most*, 1/2 for *some*, 1/4 for *hardly any*, and 0 for *none*. In case that C is ignored ($C = 1$), unweighted retention is obtained ($Retention_1$) (Lin & Hovy, 2002). In recent works (Rojas-Simón et al., 2021), the C value is calculated from the average of coverage values of each MUs found in the candidate summary (see Equation 2); n is the number of MUs found in the candidate summary.

[11] MUs refers to the sentences selected by the assessor. These sentences appear in human references.

$$C = \frac{1}{n}\sum_{i=1}^{n} Coverage(MU_i)$$

Equation 2. Average of coverage factor of $\boldsymbol{C}$.

2.4.2.2 Coverage

Coverage is defined as the degree to which one summary conveys the same information in comparison to other reference documents (Over et al., 2007; Torres-Moreno et al., 2010). This metric has been used in the first DUC workshops (DUC-2001 to DUC-2004) to manually assess automatic-generated summaries (through SEE interface), using reference documents made by humans. Initially, Coverage was considered as a fully automatic evaluation metric, but the identification of similar meanings among documents has generated complex problems that require a more specialized analysis (Over et al., 2007).

2.5 Semi-automatic evaluation

The manual evaluation supported by SEE leaded benefits because there were no standard procedures to evaluate summaries. Nevertheless, manual evaluation is time-consuming, and it requires several assessors to evaluate summaries. One way to address the issues of manual assessment is by semi-automatic evaluation. The semi-automatic evaluation considers a set of well-established features on which candidate summaries should be evaluated, some of them are mentioned in (Amigó, 2006; Nenkova et al., 2007; Nenkova & Passonneau, 2004) which are the following:

1. There is no single ideal summary because different "good" human references can be generated.
2. Employ human judgments to assess the summaries increases the cost of the evaluation.
3. Depending on the degree of understanding of the information of summary, some human references can convey better information than others.
4. The objective on which the summary is generated influences the used quality criteria.

In the state-of-the-art, the *pyramid* method has been proposed to semi-automatically evaluate the content of summaries (Nenkova et al., 2007; Nenkova & Passonneau, 2004). This method requires that humans identify similar basic units of meaning called Summary Content Units (SCUs) from a set of human references. Such SCUs are organized in a pyramidal representation that concentrates similar information from all human references. Figure 7 shows an example of a pyramid representation from four sentences obtained from different human references. The document on the left side shows that sentences are underlined in subparts whose meanings are similar. Subsequently, two SCUs are generated from the underlined portions. The first SCU (SCU1) is: *Two Libyans were officially accused of the Lockerbie bombing*, and the second SCU (SCU2) is: *The indictment of the two Lockerbie suspects was in 1991*. Each SCU has a particular weight, which corresponds to the number of summaries it appears in. Therefore, the SCU1 and SCU2 have weights equal to 4 and 3, respectively. These SCUs are organized in a pyramid model according to their weights. SCUs with higher weights are placed on the top of the pyramid because their information remains among human references. In this example, SCU1 and SCU2 are placed on the fourth (the highest) and third level of the pyramid, respectively.

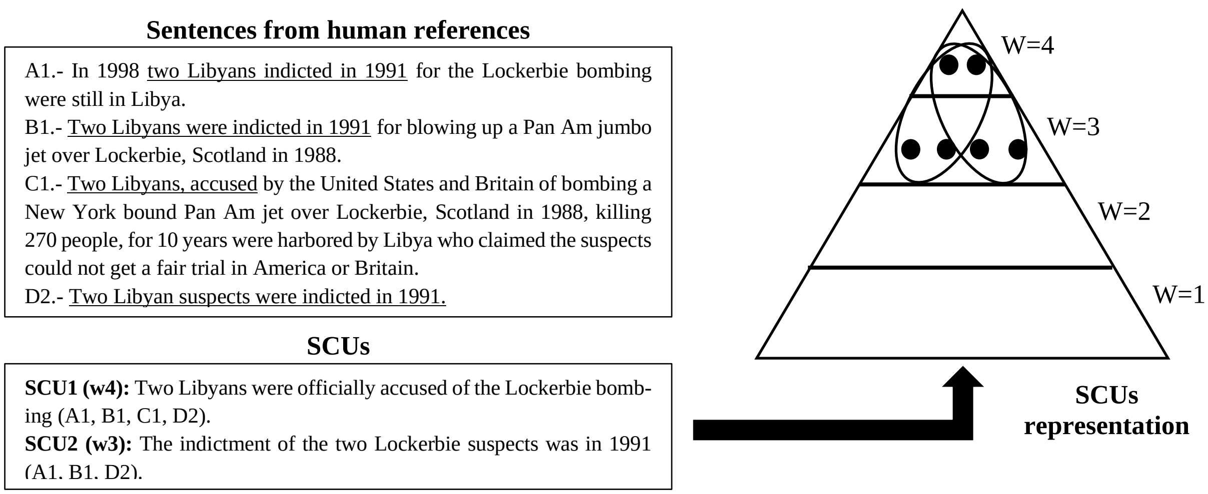


Figure 7. SCUs representation from human references (Nenkova & Passonneau, 2004).

Once obtained the pyramid of SCUs, the optimal content score of a summary is calculated through Equation 3, where n stands for the pyramid's number of levels (tiers). Thus, T_n and T_1 represent the top and bottom of the pyramid, respectively. Moreover, the weight of each SCU in tier T_i is equal to i. $|T_i|$ represents the number (cardinality) of SCUs in tier T_i. The variable j stands for the index of the lowest tier an optimally informative summary will draw from. This tier is the first one top-down such that the sum of its cardinality and the cardinalities of tiers above it is greater than or equal to X, which represents the number of SCUs found in the candidate summary.

$$Max = \sum_{i=j+1}^{n} i \times |T_i| + j \times \left(X - \sum_{i=j+1}^{n} |T_i| \right), j = max_i \left(\sum_{t=i}^{n} |T_t| \geq X \right)$$

Equation 3. Calculus of maximum pyramid score.

According to Equation 3, a summary can be considered a good one if it contains many high-level SCUs. On the other hand, if it includes more SCUs from lower levels than the higher levels, the summary has less informative content. As can be seen from the previous example, human intervention is still necessary to underline subparts will conform SCUs. The human annotation of sentences can be performed through the DUCView tool available at the following webpage: http://www1.cs.columbia.edu/~ani/DUC2005/Tool.html.

Once Max is computed from human references, we can obtain the pyramid score of each candidate summary using two evaluation formulas. The first one is named the *original pyramid score* (see Equation 4), where D_i is the number of SCUs in the summary that appears in the tier T_i and i stand for the importance weight provided to each SCU (Nenkova et al., 2007; Nenkova & Passonneau, 2004). In other words, it measures the proportion of SCUs found in the candidate summary concerning the maximum pyramid score obtained from human references (Max).

$$\mathcal{P}_{Original} = \frac{\sum_{i=1}^{n} i \times D_i}{Max}$$

Equation 4. Original Pyramid score.

Alternatively, we can evaluate under the *modified pyramid score* (see Equation 5). Unlike Equation 4, this recall-oriented formula replaces Max by the average number of SCUs obtained from the pyramid of human references. This means that theoretically, a summary can receive a score greater than 1 if it contains additional information than the average pyramid model. However, this score is not sensitive to additional information that can have a summary (Nenkova et al., 2007).

$$\mathcal{P}_{Modified} = \frac{\sum_{i=1}^{n} D_i}{\sum_{i=1}^{n} T_i}$$

Equation 5. Modified Pyramid score.

2.6 Automatic evaluation

One way to reduce the high cost of manual and semi-automatic evaluations is by automatic evaluation. Automatic evaluation allows the evaluation of many summaries without human intervention to obtain evaluated summaries in a short time. In this type of evaluation, several efforts have been carried out through the implementation of a wide variety of automated metrics and IR processes. Below, it is described the general process of automatic evaluation and some standard automatic metrics.

2.6.1 General Process of Automatic Evaluation

As mentioned in previous subsections, the ETS is a subjacent task of ATS that seeks to measure the similarity between a candidate summary and reference documents. In the state-of-the-art, several approaches have been proposed that evaluate summaries under different criteria. However, they share a general process in common. Such a process comprises four stages: *pre-processing*, *term selection*, *term weighting*, and *evaluation*. Figure 8 illustrates the stages of automatic evaluation that will be described in the next subsections.

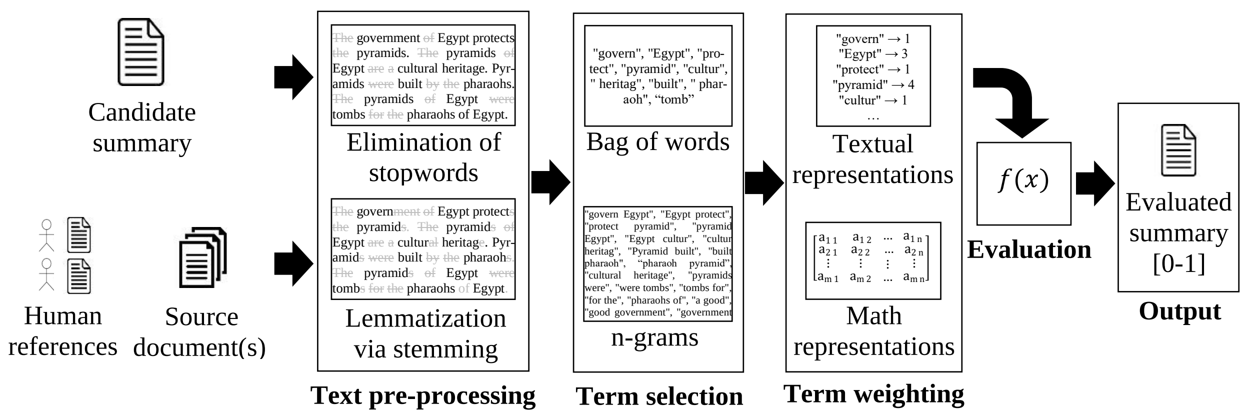


Figure 8. Stages of automatic evaluation of text summaries.

2.6.1.1 Text Pre-processing

For any NLP task, text pre-processing is an important step where the input text is filtered, eliminating unnecessary tokens that usually affect the performance of methods. According to Ledeneva & García-Hernández (2017), the election of text pre-processing techniques depends on how efficient the representation of text is. In automatic evaluation, the *elimination of stopwords* and *lemmatization via stemming* is commonly used, which are described below.

2.6.1.1.1 Elimination of Stopwords

As mentioned in previous sections, in the textual language there are different ways to express topics, concepts, definitions, or ideas in general. Therefore, all IR systems seek to "understand" such variations; but in the process, there are common words in all domains and languages that typically do not contribute to the solution because they are not good discriminants (Ledeneva & García-Hernández, 2017). Such words are known as *stopwords*. Generally, stopwords are easy to identify because they are articles, prepositions, conjunctions, verbs, adjectives, and adverbs. For several languages, stopword lists can be obtained from the Natural Language ToolKit (NLTK[12]).

In works related to Zipf's law, the most frequent words in documents are stopwords, and they are not used in many IR systems (Manning & Schutze, 1999; Zipf, 1949). Therefore, the elimination of this kind of word in documents can reduce the computational time to evaluate summaries. Moreover, it improves the degree of prediction of automatic evaluation towards manual assessment (Rojas-Simón et al., 2021; Torres-Moreno et al., 2010).

2.6.1.1.2 Lemmatization via Stemming

Another common aspect of textual language is the inflection of words. For instance, the word *organize* can derive multiple forms like *organizes*, *organized*, and *organizing*. In other cases, there are families of words with similar meanings, such as *democracy*, *democratic*, and *democratization*. Due to these situations, it is necessary to analyze words' morphological structure because those derivations usually deal with similar or the same meaning (Ledeneva & García-Hernández, 2013; Manning et al., 2009). For the ETS, it is considered good practice to reduce words' derivations to relate more terms in documents.

To capture the same meanings across the inflection of words, lemmatization techniques are often used in the state-of-the-art. In general, lemmatization refers to reduce derivations of words using vocabulary and morphological analysis (Manning et al., 2009). The most common way to perform lemmatization of words is by stemming techniques.

Stemming is the process of obtaining the root of words through the truncation of affixes, prefixes, and suffixes (Ledeneva & García-Hernández, 2017). After applying the stemming, resultant words are typically known as the *stem*. In the state-of-the-art of the ETS, stemming algorithms (especially Porter's stemming algorithm (Porter, 1980)) have been widely used to improve the degree predictability of automatic evaluation concerning manual assessment (Louis & Nenkova, 2013; Torres-Moreno et al., 2010).

[12] Stopwords lists are available at: https://bit.ly/2DqKPvW.

2.6.1.2 Term selection

Term selection is the generic process of choosing suitable text representation models that allow an effective ETS. Text representation models are techniques based on the extraction of different terms (phrases, words, or characters) of a text or document (Ledeneva & García-Hernández, 2013). For the ETS, text representation models are useful because extracted terms describe the relevance of documents. The most used text representation models are *bag of words, linear* and *skip n-grams, Longest Common Subsequences,* and *Maximal Frequent Sequences*.

2.6.1.2.1 Bag of words

The bag of words text representation model extracts all different words that appear in the text. Based on such extraction of terms, a document is represented in a set of words, where each word represents a document feature. The bag of words is considered an easy-to-use model due to the single extraction of different words (Ledeneva & García-Hernández, 2017). However, by considering a single word, polysemy increases due to the loss of context of the word itself. Moreover, this text representation model extracts too many terms, even in a small text.

For instance, from a document of five sentences (see Figure 9), the bag of words model extracts 19 terms: "The", "government", "of", "Egypt", "protects", "the", "pyramids", "are", "a", "heritage", "cultural", "were", "built", "by", "the", "pharaohs", "tombs", "for" and "good".

S1	The government of Egypt protects the pyramids
S2	The pyramids of Egypt are a cultural heritage
S3	Pyramids were built by the pharaohs
S4	The pyramids of Egypt were tombs for the pharaohs of Egypt
S5	A good government protects its cultural heritage

Figure 9. Document sample of 5 sentences of an arbitrary text.

2.6.1.2.2 N-gram

The n-gram is a text representation model that uses contiguous word sequences of a predetermined size (n) (Ledeneva & García-Hernández, 2017). The n-gram model follows the same principle as the bag of words; however, the n-gram reduces the loss of context, retrieving a more significant set of terms using n as a parameter. The bag of words model can be considered as unigrams (1-

grams). For example, if $n = 2$ (bigrams or 2-grams), the size of each retrieved term contains 2 consecutive words from the original text.

Considering the same document shown in Figure 9, the bigrams retrieved from such document are the following: "the government", "government of", "of Egypt", "Egypt protects", "protects the", "the pyramids", "pyramids of", "Egypt are", "are a", "a heritage", "cultural heritage", "pyramids were", "were tombs", "tombs for", "for the", "pharaohs of", "a good", "good government", "government protects" and "protects its".

Like the bag of words model, the n-grams do not preserve the same order that appears in the original text. Therefore, the extraction of terms with n-grams or bag of words allows the loss of information. Moreover, another inconvenience is found with the use of both models: high dimensionality. Depending on the size of the original text, retrieving terms can be a highly expensive process due to the size of each term and the number of retrieved terms.

2.6.1.2.3 Skip-grams

Another way to retrieve terms from the original text or document for ETS is by non-contiguous extraction and association of terms that consider the skip-gram text representation model (Cheng et al., 2007; Sidorov, 2019). Skip-gram is seen as a text representation model able to retrieve non-linear associations of words using arbitrary gaps (Lin, 2004). Unlike n-grams, skip-grams retrieves a broader set of terms because it considers contiguous and non-contiguous n-grams. However, skip-grams also extract n-grams with noise that makes difficult the text analysis.

For instance, considering the first sentence shown in Figure 9 (The government of Egypt protects the pyramids), the number of retrieved skip-bigrams is $C(7,2)$[13] $= 21$. Such retrieved skip-bigrams are the following: "The government", "The of", "The Egypt", "The protects", "The the", "The pyramids", "government of", "government Egypt", "government protects", "government the", "government pyramids", "of Egypt", "of protects", "of the", "of pyramids", "Egypt protects", "Egypt the", "Egypt pyramids", "protects the", "protects pyramids", "the pyramids".

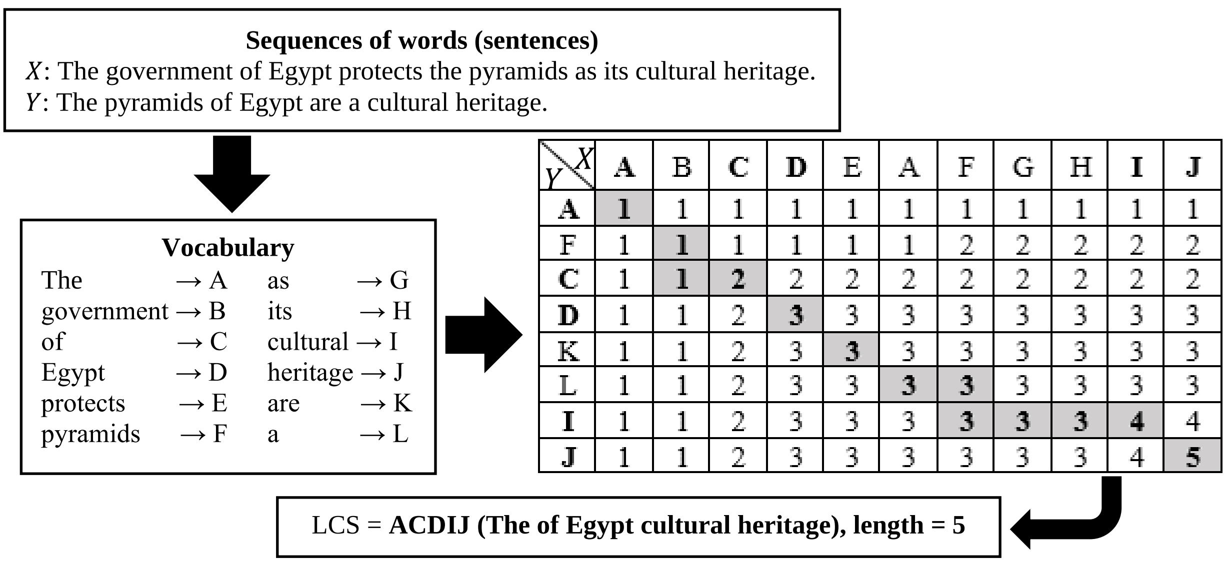
According to the previous example, skip-grams can retrieve traditional n-grams as a subset of terms. In automatic evaluation, skip-grams have been used to represent word associations that

[13] Combination: $C(7,2) = \frac{7!}{2! \times 2!} = 21$

traditional n-grams do not retrieve. Summary evaluation packages such as ROUGE uses skip-bi-grams with predefined gaps to automatically measure the quality of summaries.

2.6.1.2.4 Longest Common Subsequences

A sequence $S = (S_1, S_2, \dots, S_n)$ is a subsequence of another sequence $X = (x_1, x_2, \dots, x_m)$, if there exists a strict increasing sequence $(i_1, i_2, \dots, i_k)$ of indices of X such that all $j = 1,2, \dots, k$, there is $x_{ij} = s_j$ (Lin, 2004). Given two sequences, X and Y, the Longest Common Subsequence (LCS) of X and Y is a common subsequence with maximum length. When LCS is applied in automatic evaluation, a sentence of the summary is viewed as a sequence of words (Ledeneva & García-Hernández, 2017). For example, if two sentences (X and Y) of length m and n respectively are compared (see Figure 10), the LCS is found when there is an increasing sequence of words that sentence X and Y have in common.



Y \ X	A	B	C	D	E	A	F	G	H	I	J
A	1	1	1	1	1	1	1	1	1	1	1
F	1	1	1	1	1	1	2	2	2	2	2
C	1	1	2	2	2	2	2	2	2	2	2
D	1	1	2	3	3	3	3	3	3	3	3
K	1	1	2	3	3	3	3	3	3	3	3
L	1	1	2	3	3	3	3	3	3	3	3
I	1	1	2	3	3	3	3	3	3	4	4
J	1	1	2	3	3	3	3	3	3	4	5

Figure 10. LCS from two sentences.

According to the previous example, the longer the LCS of two sentences is, the more similar the two summaries are. In (Lin, 2004), the LCS is computed between two documents, X of length m and Y of length n, where X is a human reference sentence and Y is a candidate summary sentence.

2.6.1.2.5 Maximal Frequent Sequences

One significant shortcoming of using the bag of words and N-gram models in term selection is the order and the high dimensionality of its terms. In consequence, extracted terms would not reflect the relevant features of documents or their own extraction would be time-consuming. In view of this, Maximal Frequent Sequences (MFS) have been proposed to represent the content of texts.

A Frequent Sequence (FS) is a sequence of words or characters that appear in the text several times and in the same order. An FS is called maximal if it is not a subset of any FS (Ledeneva & García-Hernández, 2017). The MFS model finds the number of times the FS will repeat over the text to be considered frequent. This number is called *frequency threshold* (β). Figure 11 highlights five MFS with $\beta = 2$, extracted from the same document of five sentences shown in Figure 9.

S1	The **government** of Egypt **protects** the pyramids
S2	**The pyramids of Egypt** are a **cultural heritage**
S3	Pyramids were built by **the pharaohs**
S4	**The pyramids of Egypt** were tombs for **the pharaohs** of Egypt
S5	A good **government protects** its **cultural heritage**

Figure 11. MFS extracted from a document of five sentences.

Although MFS has not been used in the ETS, it has demonstrated being an adequate text representation model for improving the performance of ATS methods (Ledeneva et al., 2014). Therefore, it would be interesting and necessary to implement MFS for the ETS.

2.6.1.3 Term weighting

Term weighting is the process of assigning a numeric value for each term that reflects the degree of its relevance concerning other terms in the document (Ledeneva & García-Hernández, 2017). This process is usually performed in the ETS to enhance the association and importance of terms between the source document and the candidate summary. The most known options for term weighting are *Boolean weighting*, *Term Frequency* (TF) (Luhn, 1957), *Inverse Document Frequency* (IDF) (Salton & Buckley, 1988), and TF-IDF (Salton & Buckley, 1988). However, the development of other term weighting formulas has expanded their use in a scheme of local and global weightings (Steinberger & Ježek, 2009), as shown in Equation 6, where L_{ij} denotes the local weight for the term j of the sentence i, and G_{ij} represents the global weight for the term j in the whole document.

$$W_{ij} = L_{ij}G_{ij}$$

Equation 6. Local and global term-weighting schemes.

The following term-weighting schemes are used to assign the local weight for each term (L_{ij}):

- Frequency Weight (FW): $L_{ij} = tf_{ij}$, where tf_{ij} represents the number of times that the term j appears in the sentence i.
- Binary Weight (BW): $L_{ij} = 1$, whether the term j appears at least once in the sentence i. Otherwise $L_{ij} = 0$.
- Augmented Weight (AW): $L_{ij} = 0.5 + 0.5 \times \left(tf_{ij}/tf_{max_i}\right)$, where tf_{max_i} represents the frequency value of the most frequent term in the sentence i.
- Logarithmic Weight (LW): $L_{ij} = log\left(1 + tf_{ij}\right)$.

The following term-weighting schemes are used to assign the global weight for each j^{th} term of each i^{th} sentence (G_{ij}):

- No Weight (NW): This scheme does not establish changes for any term j ($G_{ij} = 1$).
- Inverse Sentence Frequency (ISF): $G_{ij} = log\left(N/n_j\right) + 1$, where N represents the number of sentences in the document, and n_j is the number of sentences that contain the term j.
- GFidf (GF): $G_{ij} = gf_j/sf_j$, where sf_j (sentence frequency) represents the total number of sentences in which the term j occurs, gf_j represents the total number of times that the term j is included in the whole document.
- Entropy Frequency (EF): $G_{ij} = 1 - \sum_{i=1}^{n}\left(\frac{p_{ij} \times log(p_{ij})}{log(n_{sent})}\right)$, where $p_{ij} = tf_{ij}/gf_j$ and n_{sent} is the number of sentences in the document.

Based on the local and global term weighting scheme, any document with n terms and m sentences can be represented in a matrix of $n \times m$ dimensions. For instance, Figure 12 shows an example of applying FW and GF weightings, using the document of Figure 9.

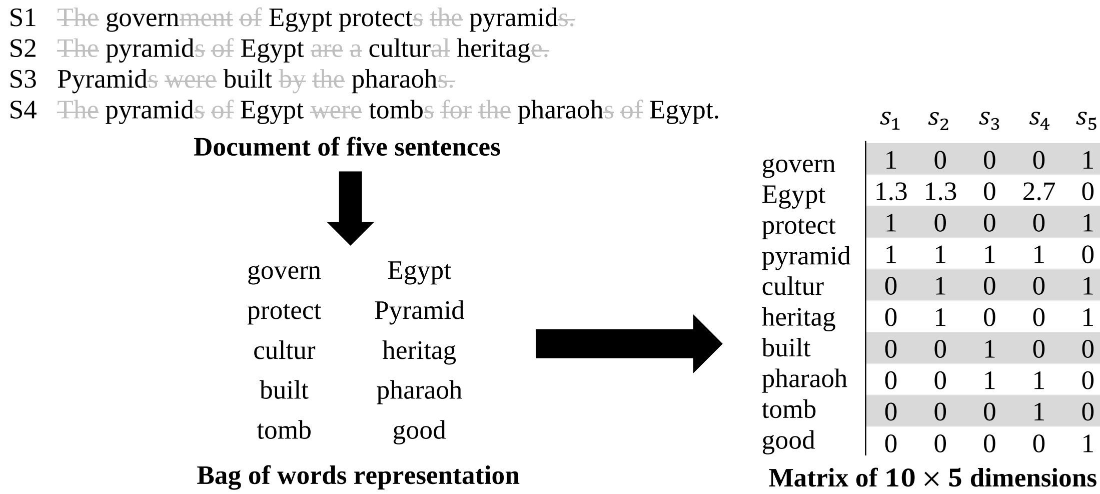


Figure 12. Example of term weighting from a document of five sentences.

From the example above, the sample document has been pre-processed through the elimination of stopwords and stemming. Such procedures are essential to retrieve and find meaningful terms from the document. Moreover, they reduce the linguistic complexity of the task. Afterward, the remaining terms are represented in a bag of words, standing for the vocabulary of the document. Based on the bag of words, term weighting is applied to generate a matrix of 10×5 dimensions. In this matrix, the term "Egypt" shows higher weight than other terms in the document. Likewise, s_4 shows higher importance than other sentences. Thus, term frequency has been a core factor in highlighting meaningful sentences and terms over the document. In the next chapter, term weighting is addressed by state-of-the-art evaluation methods, especially the LSA (see Section 3.2.2).

2.6.1.4 Evaluation

The ETS is performed under two essential criteria: (i) the utility of summaries in another given task or context (extrinsic); and (ii) the analysis of inherent features (coherence, readability, informativeness, and non-redundancy) of summaries (intrinsic). However, the automatization of the extrinsic evaluation always has been a challenging task because it requires that systems understand and measure the degree of accomplishment of summaries in different scenarios or tasks. On the other hand, extrinsic evaluation is the most studied criterion because it focuses on comparing content between the candidate summary and reference documents (source document or human references). This process requires vectorial representation and similarity of documents. Below, it is explained the idea of representing documents and calculating their similarity using the VSM and vector similarity.

2.6.1.4.1 Vector Space Model

The Vector Space Model (VSM) is the most used model for IR due to its simplicity of use in several NLP tasks where it is necessary the abstraction, representation, and comparison of text units (*e.g.*, documents, sentences, words), measuring the spatial proximity (Manning & Schutze, 1999; Salton et al., 1975; Sidorov, 2019). This model allows representing a set of features of an object in a common vector space using numerical values, which describes the importance of each feature (Manning et al., 2009; Sidorov, 2019).

The VSM is useful when it is required the representation of a set of features in high-dimensional space to describe a sentence, document, or collection. Such representation is used to perform different tasks, such as Document Classification, Document Clustering, and Text Summarization. For instance, a set of 4 words $W = \{w_1, \dots, w_4\}$ is projected in two-dimensional space (X and Y), the corresponding values of each word are: $w_1 = (0.21, 0.75)$, $w_2 = (0.42, 0.65)$, $w_3 = (0.51, 0.45)$, and $w_4 = (0.65, 0.15)$. Afterward, the vector representation is shown in Figure 13, where each word is projected by its values of features into two dimensions, presenting different inclination angles. Within the two-dimensional space, there is a spatial closeness among vectors because each one of them can be compared through similarity measures.

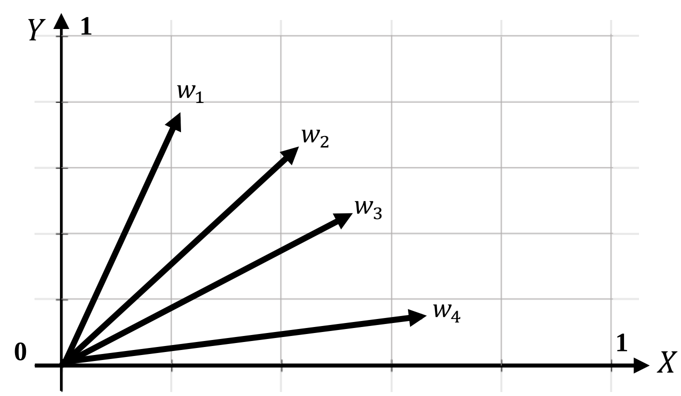


Figure 13. Vector representation of five words.

2.6.1.4.2 Vector Similarity

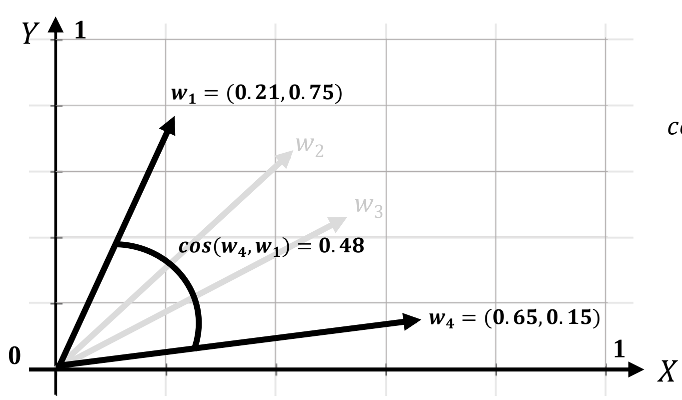
The most common manner of measuring the closeness of vectors in an n-dimensional space is by *vector similarity*. Vector similarity is a task for computation of similarity, which helps to determine the proximity of two vectors using their values of features (Sidorov et al., 2014). In several NLP tasks, vector similarity is computed through the *cosine angle* between two vectors. Typically, the calculus of the cosine angle between two given vectors is known as *cosine similarity* or *normalized correlation coefficient* (Manning & Schutze, 1999). The cosine similarity function is shown in Equation 7, where X and Y represent the vectors of features to be compared.

Like Pearson, Spearman, and Kendall correlation coefficients, cosine similarity requires a set of n values of both vectors to measure the proximity of them. Such proximity is calculated through the dot-product ($\sum_{i=1}^{n} x_i y_i$). On the other hand, normalization of vectors is considered a useful procedure when there are unfair differences in the values of features. If the vectors are normalized, cosine similarity can be calculated using the dot-product (Manning & Schutze, 1999). Otherwise, the normalization of both vectors is applied using Equation 7.

$$cos(X, Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Equation 7. Cosine similarity between two vectors.

For instance, given the vector representation of words presented in Figure 13, if w_4 and w_1 are compared by cosine similarity, then the cosine angle between w_4 and w_1 is 0.48, as shown in Figure 14. In other words, w_4 and w_1 present a vector similarity of 0.48 (in a range of 0.0 to 1.0). According to the values of features to compare, the similarity of two vectors can reach values close to 1.0, whether the values of features show proximity, or 0.0 otherwise.



$$\begin{aligned} cos(w_4, w_1) &= \frac{(0.65 \cdot 0.21) + (0.15 \cdot 0.75)}{\sqrt{0.65^2 + 0.15^2} \cdot \sqrt{0.21^2 + 0.75^2}} \\ &= \frac{0.249}{0.519} \\ &= \mathbf{0.48} \end{aligned}$$

Figure 14. Cosine angle between w_4 and w_1.

Cosine similarity is associated with automatic evaluation because the textual content of two documents is usually compared through their values of features. Explicitly, the summary evaluation by cosine similarity is performed in two steps. First, candidate summary and human references are represented in two vectors of features using the VSM. Second, the values of the features of both vectors are used to compute cosine similarity. The obtained value reflects the similarity between the candidate summary and human references. This procedure has been considered one of the first approximations in the state-of-the-art for automatic evaluation, where several summaries were automatically evaluated on large scale evaluation workshops (Radev et al., 2003).

2.6.2 Sentence-level measures

Earlier summary evaluation studies had been focused on the use of IR evaluation measures, such as *Recall*, *Precision*, and *F-measure* (Edmundson, 1969; Ledeneva & García-Hernández, 2017; Manning & Schutze, 1999), that will be described below.

2.6.2.1 Recall, Precision, and F-measure

Recall stands the proportion of correct sentences that the summarization system has forgotten (see Equation 8). If the summarization system returns a recall value equal to 1.0, then all sentences retrieved by humans are also retrieved by the summarization system.

$$Recall\ (R) = \frac{\#(Correct\ sentences)}{\#(Correct\ sentences + Unextracted\ sentences)}$$

Equation 8. Recall measure in the ETS.

Precision measures the fraction of correct sentences which the summarization system has extracted (see Equation 9). If the summarization system returns a precision value equal to 1.0, then all sentences retrieved by the summarization system are retrieved by humans.

$$Precision\ (P) = \frac{\#(Correct\ sentences)}{\#(Correct\ sentences + Incorrect\ sentences)}$$

Equation 9. Precision measure in the ETS.

F-measure stands the harmonic mean between Recall and Precision (see Equation 10). However, there are some variations where weighted values of importance are used to favors Recall or Precision (Sidorov, 2013).

$$F\text{–}measure\ (F) = \frac{2PR}{P + R}$$

Equation 10. F-measure in the ETS.

$Correct\ sentences$ are the number of sentences extracted by the summarization system and humans; $Incorrect\ sentences$ are the number of sentences extracted by the system, but not by humans; $Unextracted\ sentences$ are the number of sentences extracted by humans, but not by the summarization system (Ledeneva & García-Hernández, 2017).

2.6.2.2 Problems related to the use of sentence-level measures

IR evaluation measures are distinguished because they use the exact match of sentences between the candidate summary and human references. Initially, these measures were used to evaluate automatic extractive summaries (Donaway et al., 2000; Pardo et al., 2006; Rino et al., 2004). However, human variations have increased the complexity of human references; some of them are described as follows:

- Human references are generated in abstractive form. Several assessors of DUC and TAC workshops have made human references in an abstractive way. Generally, the abstraction of sentences is a problematical characteristic for capturing by automatic evaluation metrics, due to the semantic relationship between modified sentences. For instance, from a source document of DUC-2002 (AP880911-0016), the human reference (D061.P.100.J.B.AP880911-0016) was made by reduction, fusion, and abstraction of sentences, as shown in Figure 15.

AP880911-0016	
S1	Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.
S2	The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
S3	The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
S4	Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

D061.P.100.J.B.AP880911-0016	
S1	Tropical Storm Gilbert in the eastern Caribbean strengthened into a hurricane Saturday night.
S2	The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo.
S3	It is moving westward at 15mph with a broad area of cloudiness and heavy weather with sustained winds of 75mph gusting to 92mph.
S4	The Dominican Republic's Civil Defense alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Figure 15. Abstraction of sentences performed on DUC-2002.

- The comparison of candidate summaries with human references is considered a subjective task. Different sentences included in the candidate summary may have similar meanings concerning human references. For instance, if two candidate summaries $Sum_1 = \{5{,}9{,}15\}$ and $Sum_2 = \{5{,}9{,}11\}$ are compared with a human reference $Sum_{HumanRef} = \{5{,}9{,}11\}$, then Recall and Precision assign a better score to Sum_2 than Sum_1 because such measures use the exact match of sentences. However, sentence 15 may have the same relevance as sentence 11, but human selection only considered sentence 11 instead of sentence 15.
- The disagreement between human references decreases the accuracy of IR measures because different subsets of sentences can be selected by human selection. If two human references are made from a source document of 50 sentences (see Figure 16), then there is a disagreement between human references, and even more when the length of the source document allows several combinations of sentences.

D061.P.100.J.B.AP880911-0016	
S1	Tropical Storm Gilbert in the eastern Caribbean strengthened into a hurricane Saturday night.
S2	The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo.
S3	It is moving westward at 15mph with a broad area of cloudiness and heavy weather with sustained winds of 75mph gusting to 92mph.
S4	The Dominican Republic's Civil Defense alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

D061.P.100.J.I.AP880911-0016	
S1	Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high winds and seas.
S2	Tropical Storm Gilbert formed in the eastern Caribbean and became a hurricane on Saturday night.
S3	By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph.
S4	Flooding is expected in Puerto Rico and the Virgin Islands.
S5	The second hurricane of the season, Florence, is now over the southern United States and downgraded to a tropical storm.

Figure 16. Disagreement between human references of DUC-2002.

Addressing the issues derived from the use of IR evaluation measures, various methods based on the implementation of similarity measures (Jing et al., 1998; Radev et al., 2003; Saggion et al., 2002) and the extraction of n-grams (Lin & Hovy, 2003) were implemented in DUC workshops. In the next years, other evaluation methods, such as BLEU (Papineni et al., 2002; Pastra & Saggion, 2003), ROUGE (Lin, 2004), Basic Elements (BE) (Hovy et al., 2005, 2006), and AutoSummENG (Giannakopoulos et al., 2008) have also been proposed.

In order to tackle these issues derived from the use of IR evaluation measures, methods based on the implementation of similarity measures (Jing et al., 1998; Radev et al., 2003; Saggion et al., 2002), the extraction and statistic co-occurrence of n-grams (Lin & Hovy, 2003) were implemented in DUC workshops. In the next years, other evaluation methods such as BLEU translation evaluation (Papineni et al., 2002; Pastra & Saggion, 2003), Basic Elements (BE) (Hovy et al., 2005, 2006) and AutoSummENG (Giannakopoulos et al., 2008) have also been proposed.

2.6.3 ROUGE

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is an evaluation package for evaluating summaries automatically. Nowadays, ROUGE is the most used package due to the high correlation of their methods with the manual assessment process (Lin, 2004). The ROUGE evaluation methods assess in terms of Recall, Precision, and F-measure the statistic co-occurrence of terms between the candidate summary and human references (Ledeneva & García-Hernández, 2017). In general, ROUGE's methods use different text representation models for evaluating summaries; some of them are described below.

2.6.3.1 ROUGE-N

ROUGE-N measures the overlap of n-grams between the candidate summary and human references. ROUGE-N uses Equation 11, where n stands for the length of the n-gram ($gram_n$), and $Count_{match}(gram_n)$ represents the maximum number of n-grams that co-occurs between the candidate summary and human references (Lin, 2004). Clearly, ROUGE-N is seen as a Recall-based method because Equation 11 represents the total sum of n-grams occurring at the denominator side.

$$ROUGE{-}N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

Equation 11. ROUGE-N (Lin, 2004).

2.6.3.2 ROUGE-S and ROUGE-SU

ROUGE-S measures the statistical co-occurrence of skip-bigrams between the candidate summary and human references. Unlike ROUGE-N, ROUGE-S uses skip-bigrams to determine the quality of candidate summaries. Given the vector X of length m (candidate summary) and the vector Y of length n (human references) as two sequences of skip-bigrams, ROUGE-S uses Equation 12 to calculate the similarity of documents in terms of Recall, Precision, and F-measure as follows:

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)}, P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)}, F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2} + \beta^2 P_{skip2}}$$

Equation 12. ROUGE-S (Lin, 2004).

where $SKIP2(X,Y)$ is the number skip-bigrams that co-occurs between X and Y, β gives the relative importance between P_{skip2} and R_{skip2}; C is the combination function.

Skip-bigrams are used to evaluate summaries because it retrieves a bigger set of terms which ROUGE-N does not consider. However, ROUGE-S has the disadvantage of extracting several unnecessary skip-bigrams. Facing this problem, ROUGE-SU is proposed as an extension of ROUGE-S to evaluate summaries, considering a smaller number of terms using unigrams as a counting unit of gaps (Lin, 2004).

2.6.3.3 ROUGE-L

ROUGE-L considers the candidate summary and human references as two sequences of words. These sequences are used to find the LCS, where the resultant LCS is used to compute Recall, Precision, and F-measure, according to Equation 13.

$$R_{lcs} = \frac{LCS(X,Y)}{m}, P_{lcs} = \frac{LCS(X,Y)}{n}, F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

Equation 13. ROUGE-L (Lin, 2004).

$LCS(X,Y)$ is the length of the LCS between the candidate summary (X) and human references (Y); m and n represent the length of sequences of the candidate summary and human references, respectively. Moreover, $\beta = P_{lcs}/R_{lcs}$.

2.7 Evaluation levels

Evaluation levels are forms of comparison between automatic and manual evaluation that help to measure how similar they are. Specifically, micro and macro evaluation are well-known forms to compare the performance of automatic methods with reference to human judgments. Below, it is explained the before-mentioned evaluation levels.

2.7.1 Micro evaluation (At summary level)

According to (Ellouze et al., 2016; Louis & Nenkova, 2013), all candidate summaries are evaluated by automatic and manual evaluation. Then, the resultant scores given by automatic and manual evaluation are compared through Pearson, Spearman, and Kendall correlation coefficients. Particularly, micro evaluation is focused on how similar the automatic evaluation in comparison to manual assessment is, using evaluation scores of each candidate summary.

2.7.2 Macro evaluation (At ATS method level)

Unlike micro evaluation level, macro evaluation uses the average score of ATS methods evaluated by automatic and manual evaluation (Ellouze et al., 2016; Louis & Nenkova, 2013). Next, the resultant scores given by automatic and manual evaluation are compared through Pearson, Spearman, and Kendall correlation coefficients.

2.8 Correlation coefficients

The performance of automatic evaluation methods is determined through correlation coefficients. Correlation coefficients are used to measure the linear and non-linear relationship between automatic and manual evaluation. The primary purpose of performing this comparison is based on the assumption that a good automatic evaluation method can assign high scores to good summaries and low evaluation scores to bad summaries (Lin, 2004). In general, Pearson, Spearman, and Kendall correlation coefficients are used to measure the performance of automatic evaluation methods. Below, it is described such correlation coefficients.

2.8.1 Pearson correlation

Pearson correlation coefficient (also known as *Pearson product-moment correlation coefficient*) measures the degree of relationship between two linearly related variables (Pearson, 1895). This correlation coefficient uses Equation 14, where r represents the Pearson correlation coefficient; N represents the size of X and Y vectors; $\sum x_i y_i$ is the sum of products of each pair of values $x_i y_i$; $\sum x_i$ is the sum of values calculated from the X vector; $\sum y_i$ is the sum of values calculated from Y vector; $\sum x_i^2$ is the sum of squared x_i values, and $\sum y_i^2$ is the sum of squared y_i values.

$$r = \frac{N\sum x_i y_i - (\sum x_i \sum y_i)}{\sqrt{[N\sum x_i^2 - (\sum x_i)^2][N\sum y_i^2 - (\sum y_i)^2]}}$$

Equation 14. Pearson correlation coefficient (Pearson, 1895).

Based on this correlation coefficient, the performance of automatic evaluation is measured under the degree of its prediction towards manual assessment. Resultant values vary in the range of 1 to -1, where 1 represents a direct relationship between automatic and manual evaluation (see Figure 17a).

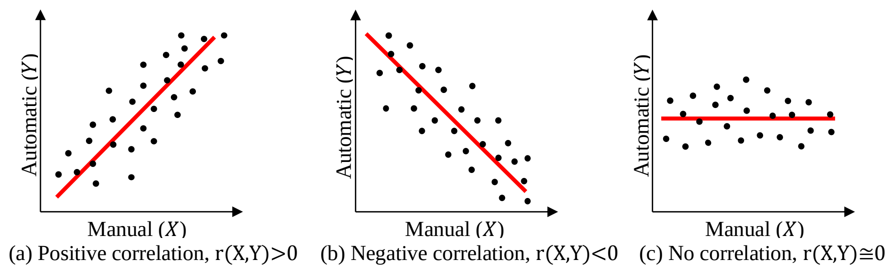


Figure 17. Examples of Pearson correlation values.

That is, automated methods assign high scores to good summaries and low scores to bad summaries. On the other hand, -1 represents an indirect (opposite) relationship of both (see Figure 17b). Finally, any value near 0 means that automatic evaluation criteria are not correlating with those of manual assessment (see Figure 17c). Therefore, there is no correlation.

2.8.2 Spearman correlation

Spearman rank correlation coefficient (also known as *Spearman rho*) measures a non-parametric association degree between two independent variables. In this correlation coefficient, the observations or values of both variables are measured on an ordinal scale to be ranked in two ordered series (Spearman, 1904). In Equation 15, Spearman rank correlation coefficient is defined, where r_s represents the Spearman correlation coefficient; d_i represents the difference between each x_i and y_i value; N represents the size of X and Y variables.

$$r_s = 1 - \frac{6 \times \sum d_i^2}{n(n^2 - 1)}$$

Equation 15. Spearman correlation coefficient (Spearman, 1904).

Unlike the Pearson correlation coefficient, the Spearman correlation assesses monotonic relationships between automatic and manual evaluation in the range of -1 to 1. Figure 18 illustrates fundamental Spearman correlation values, where the first plot (a) shows a monotonic increasing positive relationship, $r_s(X, Y) > 0$.

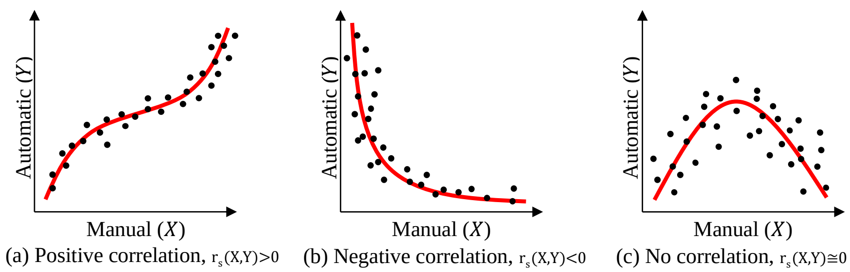


Figure 18. Examples of Spearman correlation values.

In other words, the automatic evaluation assigned high scores when the manual assessment has assigned high scores to the same summaries (micro evaluation case) or ATS systems (macro evaluation case). Under other conditions, the relation can be monotonically decreasing or negative (b) when automatic and manual evaluation have opposite directions (high scores to bad summaries and low scores to good summaries). Otherwise, relationships can be not monotonic (c) when manual assessment assigns high scores, and automatic evaluation sometimes assigns high and low scores. Therefore, there is no correlation.

2.8.3 Kendall *tau* correlation

Kendall rank correlation coefficient (also known as *Kendall tau*, τ) is a non-parametric measure that determines the association degree between two variables using a sorted vector of observations (Kendall, 1938). The observations of one variable are sorted from highest to lowest for measuring the number of agreements and disagreements between the rankings of both variables. The Kendall rank correlation coefficient is calculated from Equation 16, where S stands to the total agreement or disagreement of non-sorted observations. If the relation of each x_i and y_i value is directly proportional, then each contribution is +1. Otherwise, each contribution decrements in a rate of -1; N represents the size of X and Y variables.

$$\tau = \frac{S}{\frac{1}{2}n(n-1)}$$

Equation 16. Kendall tau correlation coefficient (Kendall, 1938).

As mentioned above, the Kendall correlation coefficient is focused on measuring the correlation of two variables under the number of concordant and discordant observations. For the ETS, it uses the number of concordant and discordant scores between automatic and manual evaluation. Figure 19 exemplifies concordant and discordant scores between manual (X) and automatic (Y) evaluation.

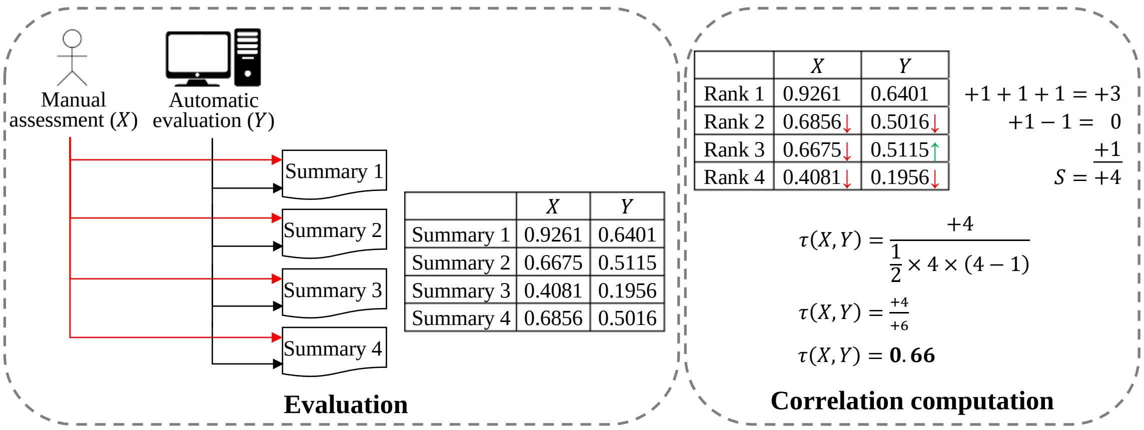


Figure 19. Example of Kendall correlation computation from manual and automatic evaluation.

In the first stage, X and Y have provided scores to four summaries in a scoring table that illustrates the automatic evaluation prediction towards manual assessment. Afterward, both sets of scores are compared through the Kendall correlation computation, where they are sorted from highest to lowest using X as a reference. This ranking intends to show concordant and discordant associations between X and Y. In the first row, there is an agreement between X and Y because both scores are the highest ones in their respective evaluations. Therefore, the contribution of this agreement is weighted to $+3$ due to the three subsequent Y scores are lower. For the next row (Rank 2), the corresponding contribution is weighted to 0 because it has a lower (0.1956) and a higher (0.5115) subsequent score. Finally, the contribution of the third row (Rank 3) is weighted to $+1$ by virtue of the following score is lower (0.1956). The overall sum of increments and decrements of contribution is equal to $+4$ ($S = +4$).

Based on the overall sum of increments and decrements, the correlation between manual and automatic evaluation equals 0.66. In other words, this result shows a moderate positive correlation that postulates the idea that there is a direct relationship between these evaluations. The Kendall correlation values vary in the range of -1 to 1, where -1 represents an indirect relationship between manual and automatic assessment and 1 represents the opposite. Moreover, when there is no relation between manual and automatic evaluation, the Kendall correlation equals 0.

2.9 Latent Semantic Analysis

Latent Semantic Analysis (LSA) (also named *Latent Semantic Indexing* or *LSI*) is a fully automatic mathematical/statistical technique taken from linear algebra that allows extracting, relating, and representing the contextual meaning of terms in a "latent" semantic space (Sidorov, 2019; Steinberger & Ježek, 2009). The LSA is useful when it is required the measurement of the contextual similarity of terms to capture the semantic information of documents (Sidorov, 2013). From the point of view of VSM, co-occurring terms are projected onto the same dimensions, or non-co-occurring terms are projected onto different dimensions (Manning & Schutze, 1999).

LSA is used in several NLP tasks, such as IR, DC, Information Filtering, and Text Summarization (Steinberger & Ježek, 2009). In automatic evaluation, the LSA is used to mathematically represent an input document (*e.g.*, candidate summary, source document, or human references) in a matrix A of $n \times m$ dimensions, where n represents the number of sentences and m represents the number of different terms used in the whole document. In other words, a matrix definition $A = [A_1, A_2, \dots, A_n]$ is presented, where each column A_i represents a vector of frequencies of each term in the sentence $i = 1, \dots, n$. The vector of frequencies is associated with term-weighting schemes (Kolda, 1998; Salton & Buckley, 1988).

For instance, given a document of five sentences (see Figure 20), the relevant terms marked in blue are selected to represent the semantic relationship among sentences in a matrix space, using the *Boolean weighting* representation. The Boolean weighting representation assigns the value of 1 whether the term appears at least once in the sentence d_i, 0 otherwise (Ledeneva & García-Hernández, 2017). The resultant representation is shown in Figure 21.

S1	The ***government*** of ***Egypt protects*** the ***pyramids***
S2	The ***pyramids*** of ***Egypt*** are a ***cultural heritage***
S3	***Pyramids*** were ***built*** by the ***pharaohs***
S4	The ***pyramids*** of ***Egypt*** were ***tombs*** for the ***pharaohs*** of ***Egypt***
S5	A ***good government protects*** its ***cultural heritage***

Figure 20. Selection of terms from document sample of 5 sentences.

The LSA defines a 10-dimensional space (whose dimensions are the 10 terms: *government*, *Egypt*, *protects*, *pyramids*, *cultural*, *heritage*, *build*, *pharaohs*, *tombs*, and *good*) and 5 objects in the space ($d_1, \ldots, d_5$). Such objects represent the number of sentences retrieved from the input document (S1, …, S5). As previously mentioned, the resultant matrix representation allows finding the association of terms from any document. However, this representation is not enough to know the importance of the terms of a document. The singular value decomposition process is needed to measure the importance of terms of any document, which is described below.

$$A = \begin{array}{l|ccccc} & d_1 & d_2 & d_3 & d_4 & d_5 \\ \hline \textit{government} & 1 & 0 & 0 & 0 & 1 \\ \textit{Egypt} & 1 & 1 & 0 & 1 & 0 \\ \textit{protects} & 1 & 0 & 0 & 0 & 1 \\ \textit{pyramids} & 1 & 1 & 1 & 1 & 0 \\ \textit{cultural} & 0 & 1 & 0 & 0 & 1 \\ \textit{heritage} & 0 & 1 & 0 & 0 & 1 \\ \textit{built} & 0 & 0 & 1 & 0 & 0 \\ \textit{pharaohs} & 0 & 0 & 1 & 1 & 0 \\ \textit{tombs} & 0 & 0 & 0 & 1 & 0 \\ \textit{good} & 0 & 0 & 0 & 0 & 1 \end{array}$$

Figure 21. LSA representation example.

2.9.1 Singular Value Decomposition

The matrix A is decomposed into three matrices using the Singular Value Decomposition (SVD). The SVD is defined in Equation 17, where $U = [u_{ij}]$ is an $m \times n$ column-orthonormal matrix, where each column is named *left singular vector*. $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a diagonal matrix of order $n \times n$, where the main diagonal includes non-negative singular values sorted in descending order. $V = [v_{ij}]$ is an orthonormal matrix of order $n \times n$, where each column is named *right singular vector*. The main idea to obtain U, Σ, and V^T matrices consist of representing the degree of importance of topics of each document.

$$A = U\Sigma V^T$$

Equation 17. Singular value decomposition.

The matrix reduction consists in modify each matrix to represent the most important r dimensions, where the order of U matrix is $m \times r$, Σ is $r \times r$, and V^T is $r \times n$, as seen in Figure 22.

Finally, the vectors retrieved from U, Σ, and V^T matrices are used to compute similarity functions through the VSM (Salton, 1989).

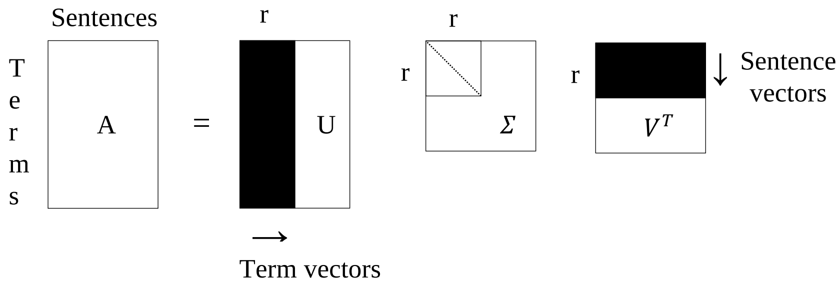


Figure 22. Dimensionality reduction for ETS (Steinberger & Ježek, 2009).

2.10 Information-theoretic measures

Another way to quantify similarity/dissimilarity between documents is by information-theoretic divergence measures. These divergence measures are based on the concept of probabilistic uncertainty or “entropy” (Cha, 2007). The entropy is defined as the average uncertainty of a single random variable $H(X) = \sum_{x \in X} p(x) log_2 p(x)$, where $p(x)$ is the probability mass function of a random variable X, over a discrete set of symbols (or alphabet) X (Manning & Schutze, 1999).

The calculus of entropy was formally defined by Claude E. Shannon in the 1940s to improve the communication signals, which are transmitted over an imperfection communication channel (Shannon, 1948). In such research, the calculus of entropy was fundamental to quantify the loss of information between two communication channels.

On the other hand, the use of entropy was not limited to measure the quality of communication signals. In ATS and ETS tasks, the entropy was fundamental to measure the loss of information between two documents, using probability distributions of terms (Lin et al., 2006). The loss of information between the two probability distributions is measured by Kullback-Leibler and Jensen-Shannon divergences. These divergence measures are described below.

2.10.1 Kullback-Leibler divergence

Kullback-Leibler divergence (also known as *Relative entropy* or D_{KL}) is defined as the average number of bits that are wasted by coding samples from a distribution p with a representation based on a distribution q (Manning & Schutze, 1999). In the context of the ETS task, D_{KL} is a measure to determine how different is the candidate summary concerning human references or source document(s). Probability distributions of terms represent such documents. Formally, D_{KL} is defined in Equation 18 (Kullback, 1978), where $p(x)$ and $q(x)$ represent the distribution probability values of two documents to compare.

$$D_{KL}(p\|q) = \sum_{x \in X} p(x) \, log_2 \frac{p(x)}{q(x)}$$

Equation 18. Kullback-Leibler divergence.

When D_{KL} is calculated from two probability distributions, then the resultant value is non-negative and $D_{KL}(p\|q) = 0$ if $p = q$. However, D_{KL} has not symmetric property $\left(D_{KL}(p\|q) \neq D_{KL}(q\|p)\right)$, does not satisfy the triangular inequality, and divergence values tend to infinity (Cover & Thomas, 2012). From the limitations of D_{KL}, other divergences measures have also been proposed, such as the Jensen-Shannon divergence.

2.10.2 Jensen-Shannon divergence

According to (Louis & Nenkova, 2009), Jensen-Shannon divergence (D_{JS}) is a measure that incorporates the idea that the distance of two probability distributions cannot be very different from the average of distances from their mean distribution. D_{JS} belongs to the set of divergence measures by the nonnegativity, finiteness, and boundedness properties (Lin, 1991). Formally, D_{JS} is defined in Equation 19, where $p(x)$ and $q(x)$ represent the distribution probability values of two documents to compare.

$$D_{JS}(p\|q) = \frac{1}{2}\left[\sum_{x \in X} p(x) log_2 \frac{2p(x)}{p(x)+q(x)} + q(x) log_2 \frac{2q(x)}{p(x)+q(x)}\right]$$

Equation 19. Jensen-Shannon divergence.

D_{JS} takes advantage of D_{KL}, using the non-negativity property. Unlike D_{KL}, D_{JS} uses the symmetric property based on the Jeffrey's divergence (*J divergence* or D_J) presented in (Jeffreys, 1946). The D_{KL} is applied twice inverting the position of both probability distributions for computing D_J ($D_J = D_{KL}(p\|q) + D_{KL}(q\|p)$). Moreover, D_{JS} follows the finiteness (non-infinity values) and boundedness properties (upper bound equal to 1.0) (Lin, 1991).

Currently, D_{KL} and D_{JS} have been used in another tasks, such as signal processing (Hershey & Olsen, 2007), pattern recognition (Ponti et al., 2017), fault detection (Chai et al., 2018; Xie et al., 2015), and text categorization (Tang & He, 2016). In automatic evaluation, the calculus of loss of information via divergence measures is considered a useful way to determine how much information the candidate summary has lost concerning human references or source document(s). In the next chapter, it is described the use of D_{KL} and D_{JS} for ETS.

2.11 Text complexity indexes

Text complexity is the inherent difficulty of reading and understanding a text (CCSSO, 2010), whose measurement depends on several factors. Some of them involve the text's readability, the text's levels of meaning or purpose, text structure, the language's conventionality, and the knowledge demands of the text (LDE, 2021). Therefore, there are different ways to measure text complexity. This section outlines the most common and used text complexity indexes in the state-of-the-art.

2.11.1 Type-Token Relationship

The Type-Token Relationship (TTR) is a simple but effective complexity index that measures the linguistic diversity of text documents (Gutierrez-Vasques & Mijangos, 2019; Kettunen, 2014). To calculate the TTR of an input document (d), we use Equation 20, where $\#Types$ are the number (cardinality) of different word types that appear in d. We can also interpret this term as *vocabulary size*. The term $\#Tokens$ represent the length (number of tokens/words) of d.

$$TTR(d) = \frac{\#Types}{\#Tokens}$$

Equation 20. TTR index.

TTR values closer than 1 indicate there is a higher diversity of words in the document, which also shows a greater complexity. Typically, the TTR index has been used in literary studies, studies in child language, and even psychiatry (Kettunen, 2014). On the other hand, it also can measure the degree of different variations of words that a language can produce (productivity).

2.11.2 Ratio of Stop Words

The most frequent words in documents (*e.g.*, *the*, *of*, *is*, *are*, and *a*) are usually considered uninformative because they do not have meaning and do not contribute to the solution of NLP tasks. Therefore, articles, prepositions, conjunctions, verbs, and adjectives are called *stop words* (Ledeneva & García-Hernández, 2017). Eliminating stop words has played an essential role in ensuring the effectiveness of proposed methods since it reduces noise and the perplexity of documents. Based on this, we describe the Ratio of Stop Words (RSW) as a complexity index that measures the overall presence of these words from an input document (d). This index is shown in Equation 21:

$$RSW(d) = \frac{StopWords(d)}{\#Tokens}$$

Equation 21. RSW index.

where $StopWords(d)$ is a function that counts the total number of stop words found in d. To perform such count, we use predefined stop word lists depending on the document's language. $\#Tokens$ represent the length (number of tokens/words) of d. RWS values near 1 indicate a high presence of these words in d, representing noise and greater complexity.

2.11.3 Ratio of Inflected Words

Word inflection is present in several languages in the world. For instance, in the English language, the word *organize* can be modified into multiple forms (*e.g.*, *organizes, organized*, *organization*, *organizational,* and *organizing*) depending on the context. Most of these modifications are characterized by adding affixes or prefixes. Therefore, the words' morphology is another feature that determines the degree of complexity in texts due to the words derivate from any inflection gain more complexity. In this sense, we define the Ratio of Inflected Words (RIW) as a complexity index that measures the proportion of inflected words from a document. The RIW index is shown in Equation 22:

$$RIW(d) = \frac{InflectedWords(d)}{\#Tokens}$$

Equation 22. RIW index.

where d represents the input document, $InflectedWords(d)$ is a function that counts the total number of words that belong to some derivation. To know whether a word is a derivation, we can use stemming algorithms, such as Porter's stemming (Porter, 1980, 2001). $\#Tokens$ represent the number of tokens or words of d. RIW values closer than 1 indicate there are many inflected words in d, which also means a greater complexity.

2.11.4 Average of Characters per Word

The Average of Characters per Word (ACW) is a complexity index that measures the mean word length of one or more documents (Ellouze et al., 2016). Formally, the ACW index is shown in Equation 23, where d represents the text of the input document, $length(w_i)$ is the number of characters of the i^{th} word (w_i), and n stands for the total number of words found in d.

$$ACW(d) = \frac{1}{n}\sum_{i=0}^{n} length(w_i)$$

Equation 23. ACW index.

This index has been widely used in readability assessment since it allows for calculating how long the words of a document can be. While longer words are included in d, the more complex the document is. However, it faces the lack of finiteness of resultant values. To solve this issue, we can normalize this equation by using the length of the longest word ($\max\left(length(w_i)\right)$), as shown in Equation 24. Thus, NACW values vary from 0 to 1, indicating the degree of proximity to the longest word.

$$NACW(d) = \frac{ACW(d)}{\max\left(length(w_i)\right)}$$

Equation 24. NACW index.

2.11.5 Average of Words per Sentence

Sentence length is a feature that describes how understandable and readable a text document can be. In this sense, the number of words per sentence may be a helpful indicator of the presence of ideas and represent more complexity. Based on this, we describe the Average of Words per Sentence (AWS) according to Equation 25:

$$AWS(d) = \frac{1}{n}\sum_{j=0}^{m} length\left(s_j\right)$$

Equation 25. AWS index.

where d represents the text of the input document, $length(s_i)$ is the number of words included in the i^{th} sentence (s_i), and m stands for the total number of sentences in d. Therefore, higher AWS values indicate longer sentences in d. Nevertheless, this index also faces the lack of finiteness in their resultant values. Due to this, we normalize this formula by using the length of the longest sentence $\max\left(length\left(s_j\right)\right)$, as shown in Equation 26.

$$NAWS(d) = \frac{AWS(d)}{\max\left(length(s_j)\right)}$$

Equation 26. NAWS index.

2.11.6 Automated Readability Index

The Automated Readability Index (ARI) was introduced by Smith and Senter (1967) to measure the degree of complexity of technical manuals, reports, staff studies, training documents, and letters of the U.S. Air Force. However, its use has been extended to other domains, such as news (Ellouze et al., 2013) and medical documents (Yan et al., 2006). The ARI is formally shown in Equation 27:

$$ARI(d) = 4.71\left(\frac{\#Characters}{\#Words}\right) + 0.5\left(\frac{\#Words}{\#Sentences}\right) - 21.43$$

Equation 27. ARI index.

where $\#Characters$ is the total number of characters in the input document (d), $\#Words$ represent the number of words, and $\#Sentences$ is the total number of sentences. As we observe, the expressions included between parenthesis also represent the ACW and AWS. The ARI values generally vary from 0 to 14, indicating the grade level needed to comprehend the text document. Table 1 lists standard ARI scores, ages, and associated grade levels to understand texts. Based on this list, we can normalize ARI scores in the range from 0 to 1, where 0 represents the score of the most understandable text and 1 (ARI≥14) is the score of those documents that are difficult to read.

Table 1. ARI score list.

Score	Age	Grade level
1	5 – 6	Kindergarten
2	6 – 7	First/second grade
3	7 – 9	Third grade
4	9 – 10	Fourth grade
5	10 – 11	Fifth grade
6	11 – 12	Sixth grade
7	12 – 13	Seventh grade
8	13 – 14	Eighth grade
9	14 – 15	Ninth grade
10	15 – 16	Tenth grade
11	16 – 17	Eleventh grade
12	17 – 18	Twelfth grade
13	18 – 24	College student
14	24+	Professor

2.11.7 Coleman-Liau Index

Unlike ARI, most readability indexes are based on counting the number of syllables or extracting Part-Of-Speech (POS) words (Feng et al., 2010). Although their use is helpful in retrieving lexical and syntactic features, these indexes are language-dependent because it is necessary to use grammar rules. Moreover, their implementation is expensive in a multilingual setting. Due to this, the Coleman-Liau Index (CLI) was proposed to gauge the understandability of texts by using the number of characters and sentences per 100 words (Coleman & Liau, 1975). The CLI is defined in Equation 28:

$$CLI(d) = (0.0588 \times L) - (0.296 \times S) - 15.8$$

Equation 28. CLI formula.

where L represents the average number of characters per 100 words and S is the average number of sentences per 100 words. The CLI is commonly used in textbooks designed for the public school system. CLI values usually vary from 0 to 17, indicating the degree of comprehension of documents. Table 2 outlines standard CLI scores and associated grade levels required to understand texts. Based on this table, we can normalize CLI scores in the range from 0 to 1, where 0 represents the score of the most understandable text and 1 (CLI≥17) is the score of the most difficult document to read.

Table 2. CLI score list.

CLI score	Grade level	Description
5 or below	Fifth grade or below	Very easy to read
6	Sixth grade	Easy to read
7	Seventh grade	Fairly easy to read
7-10	8th, 9th, and 10th grade	Conversational
11-12	11th and 12th grade	Fairly difficult to read
13-16	College	Difficult to read
17+	Professional	Extremely difficult to read

2.11.8 Word Entropy

According to Manning and Schutze (1999), entropy is a measure that calculates the average uncertainty of a single random variable. In NLP, it is a useful measure that quantifies different linguistic phenomena (Gutierrez-Vasques & Mijangos, 2019). For instance, the complexity of text documents at word and sentence levels (Conroy et al., 2010). Particularly, word entropy (H_w) is calculated according to Equation 29:

$$H_w(d) = -\frac{1}{\ln n}\sum_{i=0}^{n} p_i \ln p_i$$

Equation 29. Word entropy index.

where p_i represents the probability value of the i^{th} term in document d. The probability of each word is obtained by dividing the overall frequency of the term i and the total number of terms (n): $p_i = tf_i/n$. Notice that Equation 29 represents a normalization of Shannon's entropy (Kumar et al., 1986), such that obtained values vary from 0 and 1. H_w values near 0 indicate low entropy of words in d, and values closer than 1 represent a high variety of words.

2.11.9 Sentence Entropy

Also known as *sentence length uniformity* (Rankel et al., 2012), sentence entropy (H_s) is calculated the same way as word entropy, but it employs the sum of term probabilities of each sentence. That is, it calculates the average uncertainty of sentence lengths. Formally, the H_s index is shown in Equation 30, where $\bar{p}_j$ is the probability value of the j^{th} sentence, the letter m represents the number of sentences of document d, and n is the number of terms of each j^{th} sentence.

$$H_s(d) = -\frac{1}{\ln m}\sum_{j=0}^{m} \bar{p}_j \ln \bar{p}_j\,, \bar{p}_j = \sum_{i=0}^{n} p_i$$

Equation 30. Sentence entropy index.

This index is commonly used in ATS to penalize summaries of longer sentences (Conroy et al., 2010). But in the context of textual complexity, H_s assigns high values to documents whose sentences show a high variation of lengths. The H_s index values vary from 0 to 1, where 0 means a high uniformity of sentences and 1 represents the opposite (sentences of varying lengths).

2.12 Genetic Algorithm

The Genetic Algorithm (GA) is one of the most known evolutionary optimization techniques in the state-of-the-art, based on Darwin's natural selection principles and natural genetics to solve optimization problems (Darwin, 1859; Goldberg, 1989; Ponce, 2010). Originally known as the *genetic plan*, the GA was proposed by John Holland in 1975 as a method that seeks to simulate the actions of nature in a computer (Araujo & Cervigón, 2009; Holland, 1992). Therefore, the idea behind the GA is to do what nature does. Nowadays, it is the most popular technique in evolutionary computation research (Sivanandam & Deepa, 2008).

The GA represents through *chromosomes* the solution of a problem (Melanie, 1998). The chromosome is a simple data structure whose *genes* represent individual variables of the problem to solve. As more genes are used, the chromosome will provide a better analysis of the solution.

2.12.1 Basic Steps of the GA

According to (Melanie, 1998; Negnevitsky, 2005), the basic GA consists of the following steps:

1. Represent the variables of the problem to solve in a chromosome with a fixed length (n). Moreover, it is necessary to choose the size of chromosome population (N_{pop}), the crossover (P_c), and mutation (P_m) probabilities.
2. Define a fitness function to measure the performance of each chromosome in the population. This function is responsible for establishing the selection and generation of better chromosomes.
3. Randomly generate a population of N_{pop} chromosomes in a first generation ($g = 0$): $X_1(0), X_2(0), \dots, X_{N_{pop}}(0)$. This step is known as the *initial population step*, where each i^{th} chromosome is formed by a chain of n genes, as shown in Equation 31.

$$X_i(g) = \left[X_{i,1}(g), X_{i,2}(g), \dots, X_{i,n}(g)\right], i = 1,2, \dots, N_{pop}$$

Equation 31. Population of N_{pop} chromosomes.

4. Calculate the fitness value of each chromosome $X_i(g)$ in the population: $f\left(X_1(g)\right), f\left(X_2(g)\right), \dots, f(X_{N_{pop}}(g))$.
5. Select one couple of chromosomes ($X_{p_1}(g)$ and $X_{p_2}(g)$) from the current population as parents. Such parents are selected with a probability related to their fitness. Thus, parents with better fitness have more probabilities of being selected.
6. Apply genetic operators of crossover and mutation for the parents. In the crossover, both $X_{p_1}(g)$ and $X_{p_2}(g)$ parents are used to generate a new offspring $Y_i(g)$ with combined genes from their parents. Afterward, each chromosome $Y_i(g)$ is introduced to the mutation operator, which performs minimal genetic modifications for generating $Z_i(g)$ chromosomes. The mutation of chromosomes is based on a probability value P_m, as shown in Equation 32, where the function $Mutate\left(Y_i(g)\right)$ modifies the value of one selected gene, using a random value

(a) included in the following range: $(0–P_m]$. Otherwise, the chromosome $Y_i(g)$ is not modified.

$$Z_i(g) = \begin{cases} Mutate(Y_i(g)), & if\ a < P \\ Y_i(g), & otherwise \end{cases}$$

Equation 32. Mutation step.

7. Place the new offspring in a new population, after applying crossover and mutation genetic operators.
8. Repeat Steps 5, 6, and 7 until the new population's size becomes equal to the size of the initial population (N_{pop}).
9. Replace the initial (parent) population of chromosomes with the new population. Therefore, the population is updated according to the new offspring generated by the selection, crossover, and mutation operators. In consequence, the fitness value of the population is improved.
10. Go to Step 4 and repeat the process until the termination criterion is satisfied. The termination criterion is commonly based on a maximal number of generations, a chromosome has reached a determined fitness value, or the population has been stabilized (*i.e.*, most chromosomes share the same genes) (Sivanandam & Deepa, 2008).

Generally, the GA is seen as an iterative process whose iterations are known as *generations*, and the set of generations is called a *run* (Melanie, 1998). The number of generations is a parameter to consider because it depends on the constraints of the problem to solve. In practice, the basic GA typically iterates from 50 to 500 or more generations. On the other hand, randomness plays a fundamental role in the GA because it helps the GA explore many different solutions. Therefore, even if two GA executions are carried out with the same parameters, the GA probably will provide other solutions.

2.13 Neural networks

In the literature, there are a vast number of definitions that describe the concept of (artificial) neural networks. However, one of its most common definitions is as follows: A neural network is an interconnected assembly of several parallel processing units called *neurons* (Gurney, 1997), whose functionality resembles biological neurons' behavior (Liu & Zhou, 2020). Neurons send and receive information signals over a large number of adaptive weighted connections (Krose & van der Smagt, 1996). Such connections are important for neurons to make decisions (Heaton, 2008).

Figure 23 illustrates the general architecture of the neural network, where each node represents a neuron, and each neuron usually operates under the perceptron model.

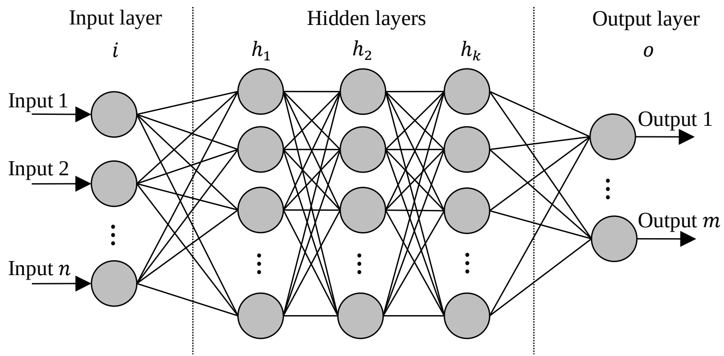


Figure 23. General architecture of the neural network.

According to the neural network architecture of Figure 23, all neurons are organized in a system of three types of layers. The first layer (input) comprises n input neurons that receive input signals for recognizing objects of n features. Afterward, input signals are introduced to a set of hidden layers that perform parallel operations to distinguish and separate the variability of signal values. Moreover, the number of hidden layers to use is variable $(1, 2, \dots, k)$ because it depends on the problem's requirements to be solved. Finally, all operations performed in hidden layers are introduced in the output layer of m neurons, which fires desirable results. Defining a neural network architecture implies determining the number of inputs, hidden, and output neurons, and the number of hidden layers.

2.13.1 Background of neural networks

Since the 50s, neural networks have been of interest in many areas of artificial intelligence and industry. Mainly, its development is related to the fact that they attempt to emulate cognitive mechanisms from the brain (Paliwal & Kumar, 2009). The origin of neural networks is inspired by observations made in the study of biological systems (Priddy & Keller, 2005). Under alternative names like *connectionist models* or *parallel distributed processing*, neural networks emerged through the biological neural study performed by McCulloch and Pitts in 1943 (Krose & van der Smagt, 1996; McCulloch & Pitts, 1943). In such a study, neurons and their components are presented as mathematical models based on biological neurons for performing computational tasks (Krose & van der Smagt, 1996).

2.13.2 Perceptron

Based on the biological neurons model, Frank Rosenblatt proposed the *perceptron* as a probabilistic model that simulates the behavior of a biological neuron to understand human capabilities of memory, generalization, recall, and thinking (Rosenblatt, 1958). Figure 24 shows the structure and organization of the perceptron.

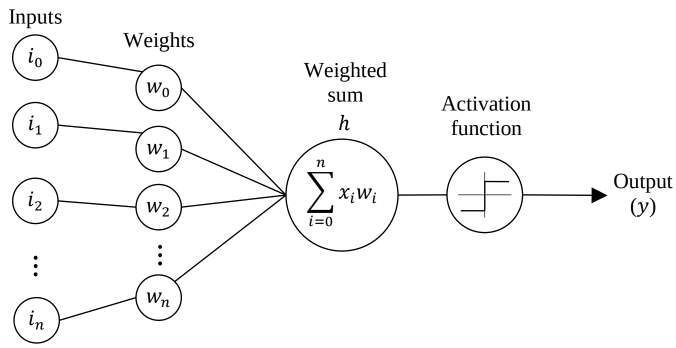


Figure 24. Structure and organization of the perceptron.

Basically, the perceptron computes a *weighted sum* of n input signals given by a vector of input features ($[x_0, x_1, \dots, x_n]$) and its corresponding weights ($[w_0, w_1, \dots, w_n]$). The result of the weighted sum (h) is introduced in an *activation function* responsible for drawing decision limits that determine the recognition and separability of objects.

In experiments related to the recognition and classification of plane patterns, the perceptron has demonstrated learning (Hay et al., 1960). However, it has one significant shortcoming. The perceptron only can learn and recognize linearly separable objects (Heaton, 2008; Minsky & Papert, 1969). In consequence, the research about neural networks was stuck because fewer efforts were carried out in the next years (Krose & van der Smagt, 1996). Notwithstanding, the interest and research of neural networks were reopened through models based on the construction of complex neural architectures (*e.g.*, feed-forward, and recurrent/feedback networks (Jain et al., 1996)).

2.13.3 Activation functions

According to (Liu & Zhou, 2020), the activation function is a kind of math operation that transforms a real number to a well-defined range $[0,1]$ or $[-1,+1]$, representing the state of activation of a neuron, where 0 or -1 indicates deactivated, and 1 indicates fully activated. The role of activation functions in neural networks is essential because they determine a non-linear separability among objects (Marques de Sá, 2001). Without these functions, neural networks would behave like simple linear functions (Sharma et al., 2017).

The interest in studying different behaviors of neural networks has led many researchers to propose several activation functions. Nowadays, there are a great variety of activation functions destined to solve various problems. The selection of activation functions depends on the problem to be solved and the desired optimization process. Below is described some activation functions.

2.13.3.1 Binary step

The *Binary Step Function* is considered the most straightforward function, which consists of transforming the value of input signals to 1 whether $h \geq \theta$ or 0 otherwise (see Equation 33), where θ represents a threshold value defined by the user.

$$y = \begin{cases} 1, & if\ h \geq \theta \\ 0, & otherwise \end{cases}$$

Equation 33. Binary step function.

For binary classification problems, this function can be considered one of the most suitable options because it establishes a simple decision boundary that activates or deactivates a neuron. However, it cannot be used in problems when a multiclass classification is required. Figure 25 illustrates the behavior of this function with $\theta = 0$ in a two-dimension plot.

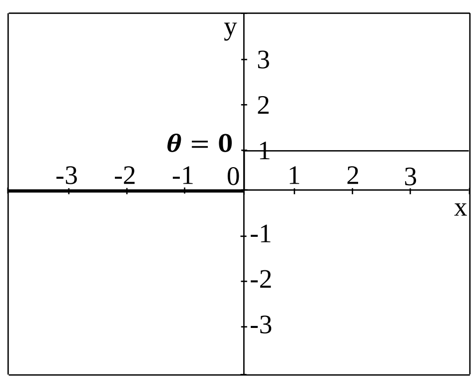


Figure 25. Binary step function when $\theta = 0$.

2.13.3.2 Linear

In general, neurons with linear activation functions act as linear approximators. Formally, the linear function is defined in Equation 34, where a represents a constant value specified by the user and h represents the weighted sum of input signals of a neuron ($\sum_{i=0}^{n} x_i w_i$) (Negnevitsky, 2005).

$$y = a \times h$$

Equation 34. Linear function.

The constant value determines the degree of inclination in the function mentioned above. For example, if $a = 1$, then the weighted sum of inputs is used as an activation function (see Figure 26). Therefore, it is necessary to define a threshold value to indicate neurons' activation or deactivation states.

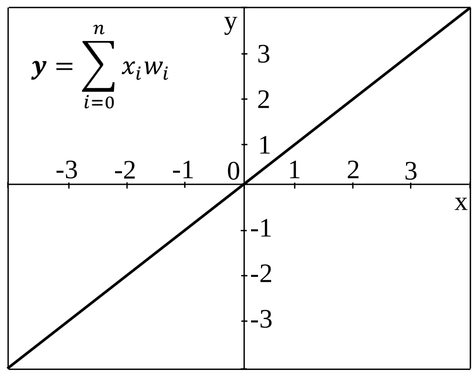


Figure 26. Linear function when $a = 1$.

2.13.3.3 Sigmoid

In many neural network architectures (especially for the multilayer perceptron network), the *Sigmoid Function* (also known as logistic function) is the most used activation function because it provides smooth continuous differentiable values in a range from 0 to 1 (see Figure 27) (Marques de Sá, 2001; Negnevitsky, 2005; Sharma et al., 2017).

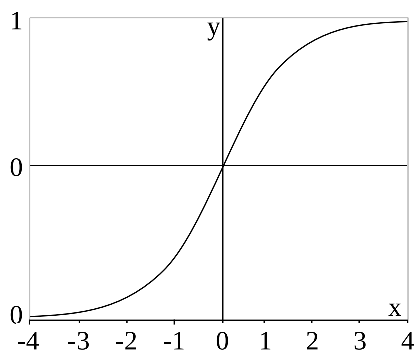


Figure 27. Sigmoid function.

Formally, the Sigmoid function is defined in Equation 35, where h represents the result of the weighted sum of n input signals of the neuron (see Figure 24). Compared to the Binary Step and

Linear functions, this function has a non-linear behavior suitable for the classification of complex patterns. Moreover, it is not too complex as other non-linear functions (Liu & Zhou, 2020).

$$y = \frac{1}{1 + e^{-h}}$$

Equation 35. Sigmoid function.

2.13.3.4 ReLU

The *Rectified Linear Unit* (*ReLU*) *activation function* is a non-linear transformation widely used in deep neural networks (Ramachandran et al., 2017). According to Sharma et al. (2017), the main advantage of using the ReLU function is that not all neurons will be activated simultaneously. Therefore, this feature allows the deactivation of a neuron only when the output is zero. The mathematical definition of the ReLU is shown in Equation 36, where h represents the weighted sum of n input signals of a neuron.

$$y = max(h, 0)$$

Equation 36. ReLU function.

Based on the above equation, the output of a neuron varies from 0 to infinity (see Figure 28). Still, when h is negative, ReLU always outputs zero, hindering the backpropagation process.

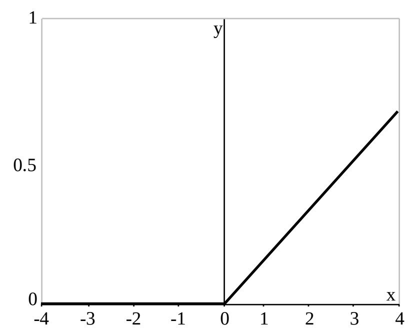


Figure 28. ReLU activation values.

2.13.3.5 Softmax

Softmax (or *Softargmax*) is an activation function based on multiple Sigmoid functions used to solve multi-class classification problems. Unlike other functions, Softmax returns a vector of K probabilities, where each real value (y_j) represents the probability of belonging to a j class, as shown in Equation 37. The variable K stands for the number of classes, the term e^{h_j} represents the standard exponential function for each j^{th} class. The term $\sum_{k=1}^{K} e^{z_k}$ performs the normalization of probabilities in the range [0,1], and the sum of all probabilities is equal to 1.

$$y_j = \frac{e^{h_j}}{\sum_{k=1}^{K} e^{z_k}}, j = 1, \dots, K$$

Equation 37. Softmax activation function.

Generally, Softmax is implemented just before the output layer with the same number of neurons as the output layer (Google, 2021). Figure 29 shows an example of the Softmax function between the last hidden and output layers. In this example, each Sigmoid is focused on classifying patterns per class. Therefore, this function is used for multiclass classification problems.

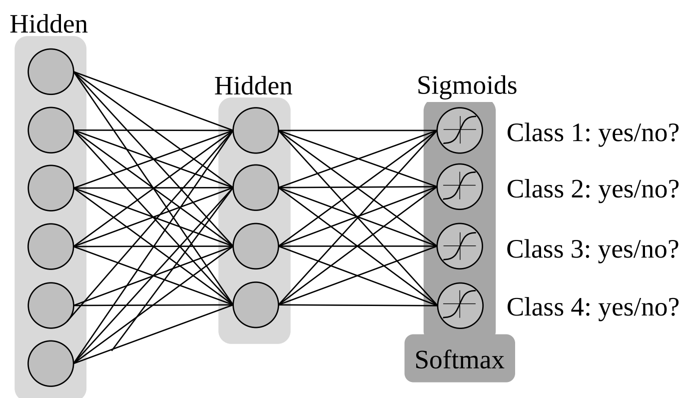


Figure 29. Softmax in a neural network.

2.13.4 Neural network architectures

In recent years, neural networks have experienced significant development because many different architectures have been proposed in the state-of-the-art. However, neural network architectures are classified into three main categories: *feedforward*, *convolutional*, and *recurrent* neural networks.

2.13.4.1 Feedforward neural network

The feed-forward neural network (FNN) is considered the most common neural network architecture (Liu & Zhou, 2020). Figure 23 exemplifies this architecture, which consists of a hierarchical structure with an input layer, several hidden layers, and an output layer of neurons. Each layer is fully connected to its neighbors. Moreover, it does not perform loops (Jain et al., 1996; Liu & Zhou, 2020). The Multilayer Perceptron (MLP) is the most common example of FNNs (Jain et al., 1996).

2.13.4.2 Convolutional neural network

The convolutional neural network (CNN) is a particular version of FNNs (Liu & Zhou, 2020), inspired by the human retina structure, whose sensory neurons have a receptive field that responds to the input stimulus. Such activity is mimicked in a neural network through a connection in the (first) hidden layer to only a small number of contiguous input neurons (Kruse et al., 2013). Unlike the traditional FNN, the CNN usually has convolutional layers, pooling layers, and several fully connected layers (Liu & Zhou, 2020). In the state-of-the-art, there are classical CNN architectures such as AlexNet (Krizhevsky et al., 2017), VGG (Visual Geometry Group) (Simonyan & Zisserman, 2014), and GoogLeNet (Szegedy et al., 2015).

2.13.4.3 Recurrent neural network

Unlike the FNNs and CNNs, the recurrent neural network (RNN) processes historical information in their neurons through loops or directed cycles (Kruse et al., 2013) that provide feedback connections (Jain et al., 1996; Liu & Zhou, 2020). In other words, an RNN has memory mechanisms to process series of data. For several tasks of NLP, the RNNs have been widely used.

2.14 Summary of the chapter

This chapter introduced several concepts to the reader about fundamental ideas of the Evaluation of Text Summaries (ETS). The first sections outlined the purpose and importance of evaluating NLP and NLG systems. Nowadays, our environment demands the creation of efficient NLG systems, so their evaluation is considered a challenging task. In the context of Text Summarization, the ETS is crucial to determine the performance of Automatic Text Summarization (ATS) methods. Most evaluation methods are classified under extrinsic and intrinsic criteria. The first one depends on how well methods measure the usefulness of summaries in specific contexts or tasks, while the second one tends to analyze the content, coherence, and informativeness of summaries.

According to the degree of human intervention, evaluation methods can also be classified into manual, semi-automatic, and automatic. For a manual evaluation, the SEE interface has been used to assess the content of summaries, obtaining content metrics, such as $Retention_w$ and Coverage. Moreover, summaries can be assessed in terms of linguistic criteria, such as grammaticality, non-redundancy, structure, and coherence. In the case of semi-automatic evaluation, the Pyramid method is the most common way to evaluate the content of summaries through a selection of Summary Content Units (SCUs). Finally, ROUGE and its variants (ROUGE-N, S, SU, and L) have been used for automatic evaluation. However, they do not work when human references are missing.

The principal aim of automatic evaluation is to imitate manual assessment. For this, we can compare them under two levels of granularity: micro and macro. The micro evaluation analyzes the proximity of automated scores to manual assessment scores at the summary level. On the other hand, macro evaluation is focused on examining the proximity of both using the average score of ATS methods. This proximity is usually measured through Pearson, Spearman, and Kendall correlations.

In literature, several strategies have been employed to evaluate summaries. Some of them implement term weighting formulas that model and represent the content of summaries. For instance, the Latent Semantic Analysis (LSA) captures the contextual meaning of words in a latent semantic space. The resultant representation requires a Singular Value Decomposition (SVD), obtaining the most information from summaries and comparing their content concerning reference documents. On the other hand, summaries can also be evaluated through entropy-based measures such as the Kullback-Leibler (D_{KL}) and Jensen-Shannon (D_{JS}) divergences.

Another aspect to take into consideration in this thesis is text complexity. Text complexity is the inherent difficulty of reading or understanding a given text, whose measurement depends on several factors, such as vocabulary diversity, quantity of information, and readability. In the state-of-the-art, text complexity indexes have been proposed, such as Type-Token Relationship (TTR), Ratio of Stop Words (RSW), Ratio of Inflected Words (RIW), Average of Characters per Word (ACW), Average of Words per Sentence (AWS), Automated Readability Index (ARI), Coleman-Liau Index (CLI), word H_w and sentence H_s entropies.

Finally, this chapter introduces the concept of two well-known methods to find solutions. The first is the Genetic Algorithm (GA), whose use is destined to obtain suitable solutions to highly complex problems through evolutionary optimization. It is inspired by Darwin's natural selection principles and natural genetics. Moreover, it is explained how the GA works through a set of 10 well-defined steps, starting with a characterization of chromosomes and the definition of parameters. Next, the user defines a fitness function to measure how well each chromosome can solve the problem. Afterward, the first generation of chromosomes is generated randomly. Then each chromosome of the population is evaluated in the fitness function. Later, genetic operators of selection, crossover, and mutation are applied to produce a new generation (offspring) of chromosomes. Finally, the offspring replaces the first generation, and genetic operators are used again until a termination criterion is satisfied (*e.g.*, number of generations). This work focuses on using the GA under the steps mentioned above, but it includes some variants that will be explained in Chapter 4.

The last method to consider in this chapter is the neural network. Neural networks are interconnected groups of several parallel processing units whose functionality resembles biological neurons' behavior. Generally, each neuron sends and receives signals over many adaptive weighted connections. That is, it operates under the perceptron model. The perceptron is the fundamental unit of neural networks because it performs mathematical operations to approximate desirable results. Moreover, it is worth mentioning that the perceptron works with activation functions that transform real numbers into well-defined values. The most common activation functions are the *Binary step*, *Linear*, *Sigmoid*, *ReLU*, and *Softmax*.

Depending on the organization of neurons (architecture), neural networks are classified into three categories: *Feedforward* (FNN), *Convolutional* (CNN), and *Recurrent* (RNN). This work focuses on using the Multilayer Perceptron (MLP) model, a particular form of FNN.

CHAPTER 3.

State-of-the-art

In recent years, several studies have been conducted to improve automatic evaluation. Some of them include the linear optimization of content metrics via the Monte Carlo method (Conroy & Dang, 2008), linear regression (Ellouze et al., 2016), and GA (Rojas-Simón et al., 2021). Others include the implementation of single and ensemble learning classifiers that combine several metrics from well-known evaluation methods (Ellouze et al., 2017a, 2017b). Although such alternatives agree with the idea of combining automatic evaluation methods through a linear or nonlinear combination of metrics, it always does not guarantee the best outcome because while we include more perspectives to solve a problem, it causes incrementing the complexity of the task. This situation is similar to other Natural Language Processing (NLP) tasks.

This chapter describes related works based on the evaluation of text summaries without human references and the selection of methods for the handwritten Text-Line Segmentation task. Moreover, ROUGE-C, SIMetrix, and LSA are described as automatic evaluation methods without human references. Moreover, the most recent methods based on the linear optimization of evaluation metrics and similarity measures are described.

3.1 Selection of handwritten Text-Line Segmentation methods

For the handwritten TLS task, this issue has been addressed by García-Calderón *et al.* (2019) through a selection of TLS methods using a complexity index called TLS-ICI (Text Line Segmentation Intrinsic Complexity Index). This index is represented in Equation 38, which is composed of an average of four normalized subindexes that measure the amount of information presented in an individual interlinear space. In other words, given an image I that represents an interlinear space from a handwritten document, we can calculate the complexity of such space using the Horizontal Projection Profile (HPP), the Vertical Projection Profile (VPP), the Histogram of Color of the Bitmap (HCB), and the Histogram of Color of the Ink (HCI).[14]

$$TLS - ICI(I) = \frac{HPP(I) + VPP(I) + HCB(I) + HCI(I)}{4}$$

Equation 38. Text-Line Segmentation Intrinsic Complexity Index (García-Calderón et al., 2019).

Once the complexity value of an interlinear space is obtained, we can estimate the overall complexity of a document according to Equation 39, where D represents the handwritten text document composed of k images of interlinear spaces (I_k^D). Therefore, the overall complexity is the average of TLS-ICI of its interlinear spaces (García-Calderón et al., 2019).

$$TLS - ICI(D) = \frac{1}{k}\sum_{i=1}^{k} TLS - ICI(I_i^D)$$

Equation 39. TLS-ICI from an image document (D).

Considering TLS-ICI calculates the degree of complexity in handwritten documents, it also provides an order of complexity (ranking) to documents. That is to say, high values are usually assigned to documents with more information, and low values to documents with lesser information. In the experimentation, such ranking was obtained from eight collections of contemporary and ancient texts written in English, Spanish, Arabic, Chinese, Greek, Khmer, Persian, Bengali, Oriya,

[14] HPP, VPP, and HCB are subindexes that measure the amount of information in interlinear spaces using entropy (Shannon, 1948). On the other hand, the subindex HCI is the average of ink from the image I (García-Calderón et al., 2019).

Kannada, and Nahuatl (Alaei et al., 2012; García-Castro, 2013; Ptak et al., 2017; Saabni et al., 2014; Sánchez-Juárez, 2016; Stamatopoulos et al., 2013; Valy et al., 2017; Villegas et al., 2016). Figure 30 shows the average TLS-ICI for each collection.

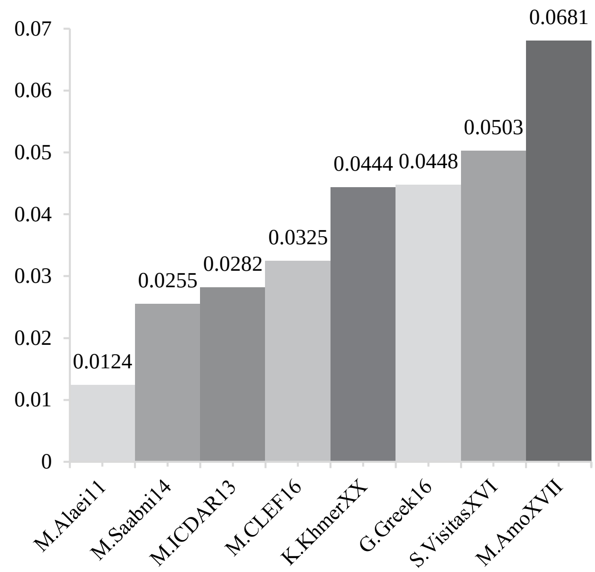


Figure 30. Ranking of average TLS-ICI for each collection of historical documents (García-Calderón et al., 2019).

As we can observe, the collection M.AmoXVII has shown the highest complexity in the ranking. Therefore, it would be expected that, in principle, the most sophisticated method performs the text line segmentation in this collection nicely. On the other side, collections of documents with lower complexities (*e.g.*, M.Alaei11 or M.Saabni14) should be analyzed by more straightforward TLS methods. Nevertheless, measuring the complexity of state-of-the-art TLS methods was also necessary. In this regard, the authors selected Ptak (Ptak et al., 2017), Arvanitopoulos (Arvanitopoulos & Susstrunk, 2014), García (García-Calderón et al., 2018), and Arivazhagan (Arivazhagan et al., 2007) methods since they are considered the best ones for the task. Afterward, the complexity of these methods was calculated by following the maximum complexity threshold with 95% of the accuracy of $MatchScore$.[15]

Figure 31A displays the results of complexity in TLS methods. The Ptak method showed the lowest complexity because it performed well in documents of lower complexities, while the

[15] $MatchScore$ is an evaluation measure that counts the number of one-to-one matches between the TLS method's detected areas and the ground truth areas (Phillips & Chhabra, 1999).

Arvanitopoulos and Arivazhagan methods obtained higher complexities. On the other hand, the García method brought the most increased complexity in this comparison, which means it is able to analyze documents with higher complexities.

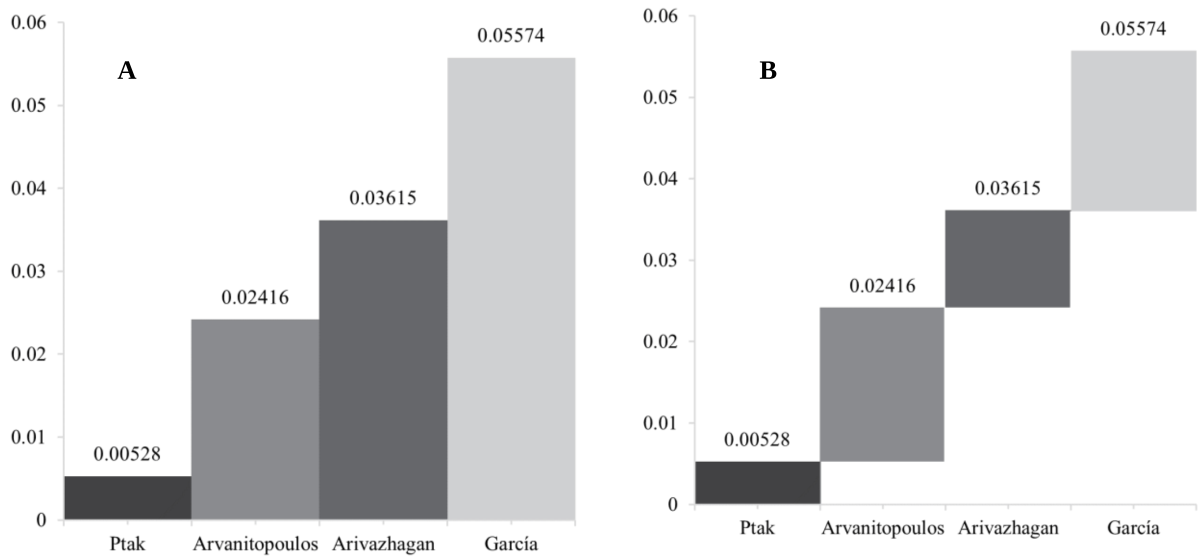


Figure 31. (A) Maximum complexity threshold for TLS methods with 95% of accuracy of MatchScore and (B) complexity values of the TLS hybrid method (García-Calderón et al., 2019).

Based on the ranking of TLS methods shown in Figure 31A, the complexity values have been used to determine the ranges that each method should be selected. Figure 31B shows the complexity ranges of Ptak, Arvanitopoulos, Arivazhagan, and García methods. As established, each technique is specialized with documents of a certain complexity. Therefore, if we consider these methods in a whole set, we also represent a Hybrid Method (HM) that chooses the most appropriate individual TLS methods. Formally, the HM is described in Equation 40:

$$HM(d) = \begin{cases} Ptak(d), & 0 \leq TLS - ICI(d) \leq 0.00528 \\ Arvanitopoulos(d), & 0.00528 \leq TLS - ICI(d) < 0.02416 \\ Arivazhagan(d), & 0.02416 \leq TLS - ICI(d) < 0.03615 \\ Garcia(d), & 0.03615 \leq TLS - ICI(d) \end{cases}$$

Equation 40. Hybrid method (García-Calderón et al., 2019).

where d means the image of the input document and $TLS - ICI(d)$ is the complexity value of d. In general, the HM method is composed of four individual TLS methods, such that only one is applied according to the complexity of the input document. So, for instance, if a document obtains a TLS-ICI value of 0.02613, it should be processed by the Arivazhagan method

($Arivazhagan(d)$). However, documents with TLS-ICI values higher than 0.03615 must be processed by the best available method ($Garcia(d)$).

3.2 Evaluation methods without human references

The evaluation of summaries with human references has been present since ATS systems experienced several advances over the last two decades. However, they present a remarkable shortcoming: they require the generation of human references that involves high costs in time and human efforts. As a result of this, the evaluation of summaries without human references has gained more attention since traditional methods only support the evaluation when human references are available. In this sense, ROUGE-C, LSA, and SIMetrix are methods focused on the comparison between the candidate summary and their source document(s) as reference documents. Below, these evaluation methods are described.

3.2.1 ROUGE-C (He et al., 2008)

As mentioned in the previous sections, ROUGE requires human references to evaluate summaries. Therefore, creating these documents is impractical, expensive, and time-consuming. Because of this, several efforts have been carried out, replacing human references with the source document(s). The first result of such efforts has been ROUGE-C (He et al., 2008), a method based on traditional ROUGE methods for evaluating summaries without human references.

Typically, traditional ROUGE methods are known for receiving human references as model documents (*model* in short) and the candidate summary as a test document (*test* in short) to evaluate summaries (see Figure 32). Instead of this, ROUGE-C receives input documents in inverse order, where the source document is used as a *test*, and the candidate summary is used as a *model*.

The configuration proposed by He et al. (2008) intends to change the order of input documents for determining co-occurring terms of them. In this way, traditional ROUGE methods, such as ROUGE-N, L, W, S, and SU4 can evaluate without human references, generating the following variants: ROUGE-C-N, L, W, S, and SU4.

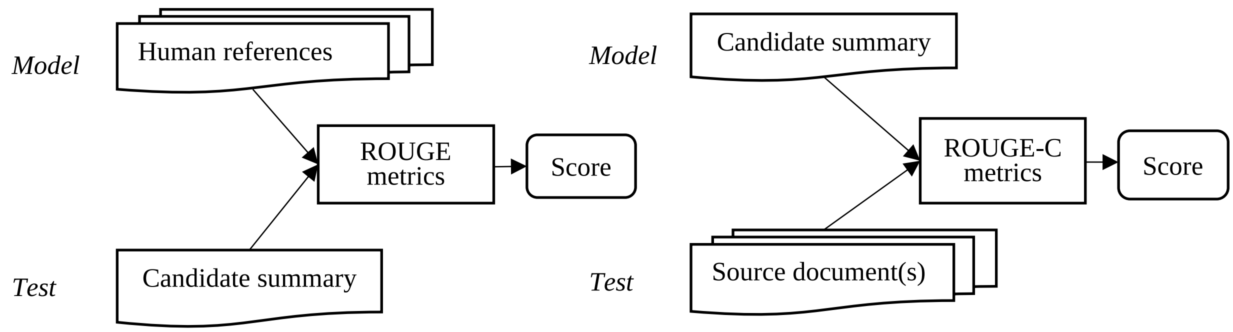


Figure 32. Main difference between ROUGE and ROUGE-C (He et al., 2008).

For instance, the evaluation of ROUGE-N without human references is called ROUGE-C-N. Formally, ROUGE-C-N is defined in Equation 41, where n represents the length of n-grams ($gram_n$), $Count_{match}(gram_n)$ is the maximum number of n-grams that co-occurs between the candidate summary and the source document.

$$ROUGE\text{–}C\text{–}N = \frac{\sum_{S \in \{PeerSummary\}} \sum_{gram_m \in S} Count_{match}(gram_n)}{\sum_{S \in \{SourceDocument\}} \sum_{gram_n \in S} Count(gram_n)}$$

Equation 41. ROUGE-C-N (He et al., 2008).

In summary, ROUGE-C-N measures the ratio of n-grams between the candidate summary and its source document(s). Therefore, ROUGE-C-N is considered as a precision-based method, because of the denominator of Equation 41 shows the total number of n-grams obtained from the source document. The ROUGE-C's metrics were evaluated through human judgments given by $Retention_w$ and Coverage from DUC01, DUC02, and DUC03 datasets.

3.2.2 LSA for ETS (Steinberger & Ježek, 2009)

Steinberger & Ježek (2009) proposed using LSA (described in Section 2.9) to evaluate summaries without human references. The underlying idea of using LSA is that this method can capture the topics from the candidate summary to compare them to the main topics from the source document. To achieve this, the authors of this work suggest using LSA under the following steps:

1. Matrix representation of input documents
2. Decomposition of values through the SVD
3. Comparison of resultant values using the VSM

The first step consists of representing the candidate summary and the source document into two matrices A of $m \times n$ dimensions (see example of Figure 21), where n and m represent the number of sentences and the number of terms obtained from the document, respectively. Both matrices are obtained through the combination of local and global term-weighting schemes proposed in (Chisholm & Kolda, 1999; Salton & Buckley, 1988). The calculus of each value a_{ij} of each matrix is based on Equation 42, where L_{ij} denotes the local weight for the term j of the sentence i, and G_{ij} represents the global weight for the term j in the whole document.[16] Figure 12 illustrates a practical example of a matrix representation from a sample document of five sentences using the FW and GF weightings. Notice the sample document has been pre-processed by eliminating stop-words and performing stemming.

$$a_{ij} = L_{ij} \times G_{ij}$$

Equation 42. Generation of input SVD matrix.

For the second step, the candidate summary (A) and the source document (A') matrices are decomposed into three matrices through the SVD (see Equation 43). Each matrix represents the degree of importance of input documents. The matrices U and U' represent the degree of importance of terms, Σ and Σ' represent the singular values of each input document, and V^T and V'^T represent the degree of importance of sentences.

$$A = U\Sigma V^T, A' = U'\Sigma' V'^T$$

Equation 43. Decomposition of values from A and A'.

Moreover, the SVD considers a dimensionality reduction to show the most important topics of each document. That is, each matrix is reduced in r dimensions, where the dimension of U is $m \times r$, Σ is $r \times r$, and V^T is $r \times n$, as seen in Figure 22. In this way, the candidate summary and its source document show their main topics' degree of importance.

Finally, the matrices of both documents are compared in the third step. The comparison of resultant values given by the SVD is performed in two ways: Main Topic Similarity (MTS) and Term Significance Similarity (TSS). MTS is a procedure that extracts the first column of matrix U and U' to compare them via cosine similarity, as shown in Equation 44.

[16] Local and global term-weighting formulas are described in Section 2.6.1.3.

$$Sim_{cos}(U,U') = \frac{\sum_{i=1}^{N} u_i \times u'_i}{\sqrt{\sum_{i=1}^{N} u_i^2} \times \sqrt{\sum_{i=1}^{N} u'^2_i}}$$

Equation 44. Cosine similarity between U and U'.

On the other hand, TSS requires a dimensionality reduction to determine the most relevant topics of both documents. Therefore, the TSS uses the matrices U, U', Σ, and Σ' to calculate the degree of importance of the terms. Subsequently, the most relevant topics are compared, generating a resultant score that indicates the ratio of topics that the candidate summary has preserved concerning its human references. In summary, the general process of evaluation with the LSA is led according to Figure 33.

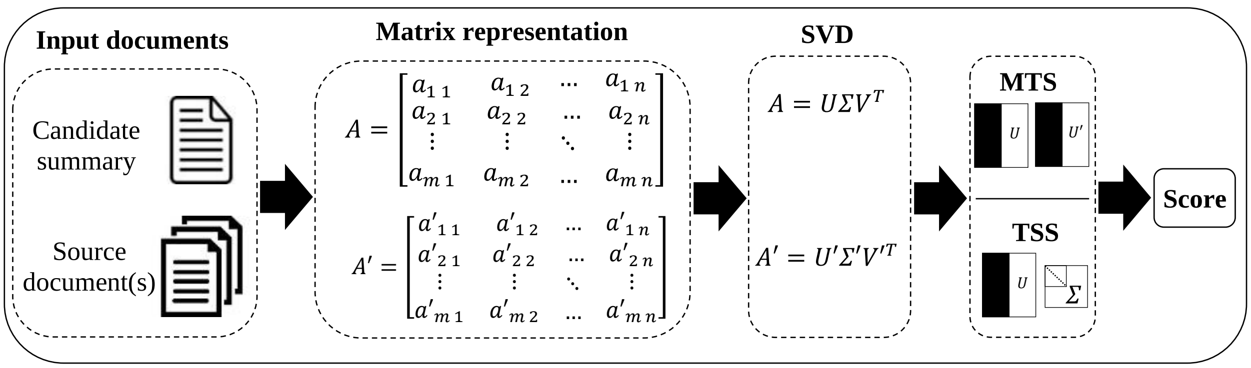


Figure 33. Evaluation of summaries without human references via LSA.

3.2.3 SIMetrix (Louis & Nenkova, 2013)

SIMetrix[17] (Summary Input similarity Metrics) is an evaluation package based on 10 similarity/distance metrics and measures to evaluate summaries without human references. Some of them are based on vector similarity, summary likelihood, and topic signature words (Louis & Nenkova, 2013; Rojas-Simon et al., 2022). However, the metrics with the best performance derive from divergence measures, such as D_{KL} and D_{JS}. The D_{KL} treats the candidate summary and its source document as two probability distributions to calculate their differences. Formally, D_{KL} is defined in Equation 45, where P_w represents the probability distribution of the term w in the source document; Q_w represents the probability distribution of the term w in the candidate summary.

[17] SIMetrix can be downloaded from https://github.com/igorbrigadir/simetrix

$$D_{KL}(P\|Q) = \frac{1}{2}\sum_{w} P_w log_2 \frac{P_w}{Q_w}$$

Equation 45. Kullback-Leibler divergence for ETS.

Although this measure has been a suitable option for evaluating summaries, it does not have the symmetric property or the triangular inequality. Moreover, D_{KL} values tend to infinity. Addressing these types of problems, D_{JS} was implemented in SIMetrix to measure the difference between the candidate summary and its source document. Formally, D_{JS} is defined in Equation 46, where P_w represents the probability distribution of the term w in the source document; Q_w is the probability distribution of the term w in the candidate summary.

$$D_{JS}(P\|Q) = \frac{1}{2}\left[\sum_{w} P_w log_2 \frac{2P_w}{P_w + Q_w} + Q_w log_2 \frac{2Q_w}{P_w + Q_w}\right]$$

Equation 46. Jensen-Shannon divergence for ETS.

In particular, the probability distribution of each term w is calculated through Equation 47, where P_w stands for the ratio between the number of occurrences of the term w in the source document or SD (C_w^{SD}) and the number of terms obtained from the source document and the candidate summary ($N = n_{CS} + n_{SD}$).

$$P_w = \frac{C_w^{SD}}{N}, Q_w = \begin{cases} \dfrac{C_w^{CS}}{n_{CS}}, & if\ w \in CS \\ \dfrac{C_w^{SD} + \delta}{N + \delta * B}, & otherwise \end{cases}$$

Equation 47. Distribution of terms formulas.

On the other hand, Q_w stands for the ratio between the number of occurrences of the term w in the candidate summary or CS (C_w^{CS}) and the sum of terms retrieved from the candidate summary (n_{CS}), whether each term w appears in the candidate summary. Otherwise, it requires smoothing parameters, where $\delta = 0.005$ and $B = 1.5|V|$. The constant V represents the number of different terms retrieved from the candidate summary.

Initially, D_{KL} and D_{JS} were proposed to measure the loss of information between two communication signals (Kullback, 1978; Kullback & Leibler, 1951; Lin, 1991). However, these divergence measures have been adapted to measure how much information has lost the candidate summary in comparison its source document(s). In general, SIMetrix's divergence measures were evaluated from scores provided by the Pyramid method on the TAC08 dataset.

In view of obtaining high correlation scores with divergence measures, other evaluation packages like FRESA (FRamework for Evaluating Summaries Automatically) and SummTriver have incorporated these measures to evaluate summaries without human references (Cabrera-Diego & Torres-Moreno, 2018; Torres-Moreno et al., 2010).

3.3 Evaluation methods based on the linear combination of measures

Another way to evaluate summaries is by the linear combination of automatic measures. In the state-of-the-art, the combination of metrics is considered a convenient alternative that improves the performance of automatic evaluation (Ellouze et al., 2016). Furthermore, recent studies have demonstrated that the linear optimization of measures can combine different evaluation criteria, reducing the gap between automatic and manual evaluation (Conroy & Dang, 2008; Ellouze et al., 2013, 2016; Rojas-Simón et al., 2021). Based on this, canonical correlation analysis, linear regression, and SECO-SEVA have been presented as two alternatives to improve the performance of automatic evaluation. This section briefly describes both alternatives in depth.

3.3.1 ROSE (Conroy & Dang, 2008)

In (Conroy & Dang, 2008), the authors proposed ROUGE Optimal Summarization Evaluation (ROSE) as an evaluation method based on the linear combination of seven well-known metrics from the ROUGE package (ROUGE-1, 2, 3, 4, L, and SU4) and Basic Elements (BE). This method was proposed under the assumption that different variants of ROUGE may capture several qualities on which the human expert evaluates a summary. For instance, ROUGE-1 extracts unigrams (words) as a text representation model to evaluate summaries, but longer variants of ROUGE can capture linguistic qualities that ROUGE-1 does not consider.

ROSE applies the principles of Canonical Correlation Analysis (CCA) (Härdle & Simar, 2015; Hotelling, 1935) through the Monte Carlo method (Robert & Casella, 2004) to obtain several combinations of n evaluation measures. Therefore, the best combination seeks to correlate better with human judgments. Formally, ROSE is represented in Equation 48, where a_{ij} represents the evaluation score of the metric j for each i^{th} summary ($i = 1, \dots, m$), and the variable w_j represents the relevance factor of each j^{th} evaluation measure. The obtained result of this combination is compared to manual assessment scores (b_i) via the Pearson correlation coefficient.

$$x = argmax\ r\left(\sum_{j=1}^{n} a_{ij}\, w_j, b_i\right)$$

Equation 48. Canonical correlation analysis (Hotelling, 1935).

Experimentally, ROSE was used to evaluate summaries on DUC05, DUC06, and DUC07 datasets, and it was correlated to Responsiveness (a human assessment measure). The performance of ROSE was evaluated under a technique of four-fold cross-validation. In other words, the authors split 3/4 of the evaluated summaries from one dataset to build the ROSE predictive model and 1/4 from another dataset for testing it. However, the main disadvantage of ROSE is the random weighting of evaluation measures given by the Monte Carlo method, where the provided solutions show uncertainty in proximity toward human assessments.

3.3.2 Linear regression for ETS (Ellouze et al., 2016)

Ellouze et al. (2016) proposed a linear combination of 21 linguistic and content-based evaluation measures. On the one hand, the content-based measures were obtained from ROUGE (Lin, 2004), BE (Hovy et al., 2006), and AutoSummENG packages (Giannakopoulos et al., 2008; Giannakopoulos & Karkaletsis, 2011). On the other hand, linguistic-based measures have also been considered, such as Gunning Fog Index, Flesch Reading Ease, Flesch-Kincaid Index, and others (Feng et al., 2010). However, these measures showed lower relevance than ROUGE and AutoSummENG.

In the context of ETS, Linear Regression uses an X vector of length $n+1$ to maximize the correlation of automatic evaluation concerning manual assessment (Ellouze et al., 2013). Therefore, the linear regression is shown in Equation 49:

$$x = argmax\ r\left(w_0 + \sum_{j=1}^{n} a_{ij}\, w_j, b_i\right)$$

Equation 49. Linear combination of evaluation metrics with linear regression.

where w_0 represents the intercept value, a_{ij} represents the j^{th} evaluation score to the summary i (micro-evaluation case) or ATS system i (macro-evaluation case); w_j represents the relevance value for each j^{th} evaluation score. On the other hand, b_i represents the Pyramid evaluation score for each summary i (micro-evaluation case) or ATS system i (macro-evaluation case). The result of the linear combination ($w_0 + \sum_{j=1}^{n} a_{ij}\, w_j$) and the pyramid scores (b_i) are compared via Pearson correlation coefficient.

Unlike ROSE, most of these metrics are based on different evaluation approaches to improve the correlation of automatic evaluation. The performance of this method was analyzed on the TAC08 dataset, measuring its proximity concerning scores given by the Pyramid method (Nenkova & Passonneau, 2004). However, the linear regression of metrics does not establish a normalized range of evaluation scores. Therefore, the lower and upper bounds could not be well-identified. On the contrary, ROUGE's measures establish as the lower bound a value equal to 0.0 and the upper bound equal to 1.0, representing the lowest and the highest quality, respectively.

3.3.3 SECO-SEVA (Rojas-Simón et al., 2021)

SECO-SEVA stands for a SElf-COntent Summary EVAluation (Rojas-Simón et al., 2021).[18] That is, it is an evaluation method that does not require human references to evaluate summaries because it compares the content of summaries with their corresponding source document(s). In addition, SECO-SEVA is based on the linear optimization of 31 content measures derived from the LSA, SIMetrix, and ROUGE-C methods via Genetic Algorithm (GA). Equation 50 formally describes SECO-SEVA,

$$SECO\text{–}SEVA\left(D_{Smry}, D_{SrcDocs}\right) = \sum_{i=1}^{31} w_i a_i\left(D_{Smry}, D_{SrcDocs}\right)$$

Equation 50. SECO-SEVA (Rojas-Simón et al., 2021).

where D_{Smry} and $D_{SrcDocs}$ represent the candidate summary and source documents content, a_i depicts the evaluation of the i^{th} measure, and w_i stands for the relevance value associated with each i^{th} measure in the range $[0-1]$. Table 3 shows the list of 31 evaluation measures used in this study. As observed, seven measures come from ROUGE-C, 16 are term-weighting variations of the LSA method, and the last eight are smoothed and non-smoothed variants of the D_{JS}.

[18] SECO-SEVA is an evaluation package available at the following GitLab repository: https://gitlab.com/JohnRojas/SECO-SEVA.

Table 3. List of 31 content evaluation measures used in (Rojas-Simón et al., 2021).

ROUGE-C	LSA		SIMetrix
ROUGE-C-1	$LSA(NW, FW)$	$LSA(NW, FW)$	JS_1
ROUGE-C-2	$LSA(NW, AW)$	$LSA(NW, LW)$	JS_2
ROUGE-C-3	$LSA(ISF, FW)$	$LSA(ISF, BW)$	JS_3
ROUGE-C-4	$LSA(ISF, AW)$	$LSA(ISF, LW)$	JS_4
ROUGE-C-5	$LSA(GF, FW)$	$LSA(GF, BW)$	$JS\text{–}Smth_1$
ROUGE-C-L	$LSA(GF, AW)$	$LSA(GF, LW)$	$JS\text{–}Smth_2$
ROUGE-C-SU4	$LSA(EF, FW)$	$LSA(EF, BW)$	$JS\text{–}Smth_3$
	$LSA(EF, AW)$	$LSA(EF, LW)$	$JS\text{–}Smth_4$

FW: Frequency Weight, BW: Binary Weight, AW: Augmented Weight, LW: Logarithmic Weight, NW: No Weight, ISF: Inverse Sentence Frequency, GF: GFidf, and EF: Entropy Frequency. These representations are obtained according to the formulas presented in Section 2.6.1.3.

As we can expect, some measures are more relevant than others because they show more affinity to human judgments. For instance, Figure 34 displays the ranking of the relevance of 31 evaluation measures on the micro-evaluation level of the DUC01 dataset. Notice that ROUGE-C-L showed the highest relevance in the ranking (0.54041). In other words, ROUGE-C-L offers 54.041% of affinity to human judgments. On the other hand, the remaining 30 measures contribute 0.45959 (45.959%) to SECO-SEVA.

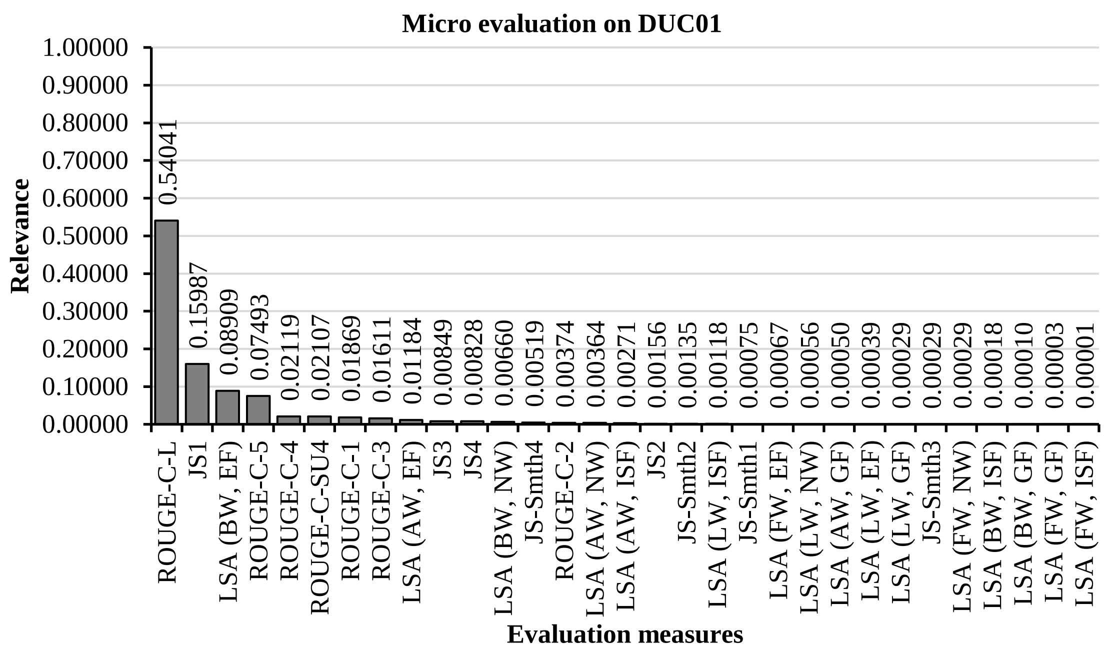


Figure 34. Relevance of 31 measures on micro evaluation level of DUC01 (Rojas-Simón et al., 2021).

The performance of SECO-SEVA was tested according to its degree of proximity toward human judgment scores on DUC01 and DUC02 datasets using Pearson, Spearman, and Kendall correlation coefficients (Rojas-Simon et al., 2022). Moreover, obtained correlation results were compared to state-of-the-art evaluation methods (see Table 4). This comparison enables the analysis of the strengths or weaknesses of all evaluation measures under specific evaluation criteria.

Table 4. Comparison between SECO-SEVA and state-of-the-art measures on micro evaluation level.

Metrics	DUC01 ($Retention_w$)			DUC02 (Coverage)		
	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall
SECO-SEVA	0.46735 (3)	0.42968 (3)	0.30012 (3)	0.64394 (1)	0.60967 (2)	0.44364 (1)
ROUGE-C-2	0.44499 (9)	0.40609 (8)	0.28256 (8)	0.61271 (8)	0.60650 (4)	0.44015 (4)
ROUGE-C-3	0.44388 (10)	0.40571 (9)	0.28229 (9)	0.61089 (10)	0.60647 (5)	0.43995 (5)
ROUGE-C-SU4	0.44678 (8)	0.41110 (7)	0.28618 (7)	0.61291 (7)	0.60847 (3)	0.44159 (3)
$LSA(AW, NW)$	0.45454 (5)	0.41748 (5)	0.29042 (5)	0.63737 (4)	0.60116 (7)	0.43573 (7)
$LSA(AW, GF)$	0.45195 (6)	0.41299 (6)	0.28722 (6)	0.63759 (3)	0.60017 (8)	0.43479 (8)
$LSA(AW, EF)$	0.45946 (4)	0.42337 (4)	0.29473 (4)	0.64020 (2)	0.60515 (6)	0.43896 (6)
JS_1	0.47534 (1)	0.44154 (1)	0.30827 (1)	0.61197 (9)	0.61037 (1)	0.44350 (2)
JS_2	0.45127 (7)	0.40524 (10)	0.28227 (10)	0.61663 (5)	0.59892 (10)	0.43411 (9)
JS_3	0.43763 (12)	0.39191 (11)	0.27260 (11)	0.61590 (6)	0.59566 (11)	0.43142 (11)
$JS\text{–}Smth_1$	0.47393 (2)	0.44136 (2)	0.30790 (2)	0.53948 (11)	0.59931 (9)	0.43286 (10)
$JS\text{–}Smth_2$	0.43886 (11)	0.38855 (12)	0.26996 (12)	0.49808 (12)	0.51101 (12)	0.36151 (12)
$JS\text{–}Smth_3$	0.40559 (13)	0.35447 (13)	0.24535 (13)	0.48808 (13)	0.48113 (13)	0.33904 (13)

Note: The number in parentheses in each table slot shows the ranking of each measure in a specific dataset and correlation coefficient.

According to the correlation results shown in Table 4, SECO-SEVA appears in the first position of each comparison. Especially, it obtained the highest correlations on the DUC02 dataset. That is, SECO-SEVA correlates best with Coverage than state-of-the-art evaluation measures. However, it is surpassed by JS_1 and $JS{-}Smth_1$ on the DUC01 dataset, but it remains one of the best evaluation methods. This is because perhaps some evaluation criteria may differ between $Retention_w$ and Coverage. For this reason, this thesis proposes an automatic selection of content-based measures according to the text complexity of source documents.

3.4 Summary of the chapter

This chapter was divided into three parts to know the state-of-the-art methods and techniques related to this thesis. Table 5 provides a general summary of all the state-of-the-art methods and techniques seen in this chapter.

Table 5. Summary of state-of-the-art methods and techniques.

Topic	Method	Task	Technique used	Text representation model	Are human references required?	Manual evaluation measure	Corpus / dataset
3.1 Selection of handwritten TLS methods	3.1.1 TLS-ICI (García-Calderón et al., 2019)	Handwritten TLS	Hybrid TLS method	N/A	Yes	N/A	M.Alaei11 M.Saannil4 M.ICDAR13 M.CLEF16 K.KhmerXX G.Greek S.VisitasXVI M.AmoXVII
3.2 Evaluation methods without human references	3.2.1 ROUGE-C (He et al., 2008)	ETS	Statistic co-occurrence of terms	Bag of words N-grams Skip-grams LCS	No	$Retention_w$ Coverage	DUC01 DUC02 DUC03
	3.2.2 LSA (Steinberger & Ježek, 2009)	ETS	LSA	Term-weighting schemes	No	Coverage	DUC02
	3.2.3 SIMetrix (Louis & Nenkova, 2013)	ETS	Loss of information via divergence measures	N-grams	No	Pyramids	TAC08
3.3 Evaluation methods based on linear combination of measures	3.3.1 ROSE (Conroy & Dang, 2008)	ETS	CCA via Monte Carlo	Combination of seven ROUGE's measures	Yes	Responsiveness Readability	DUC05 DUC06 DUC07
	3.3.2 Linear regression (Ellouze et al., 2016)	ETS	Linear regression	Combination of 21 measures	No	Pyramids	TAC08
	3.3.3 SECO-SEVA (Rojas-Simón et al., 2021)	ETS	Linear optimization via GA	Optimization of 31 content measures	No	$Retention_w$ Coverage	DUC01 DUC02

N/A = Not Applicable

The first section of this chapter presented a related work that proposes a selection of automatic TextLine Segmentation (TLS) methods for the handwritten TLS task. The authors proposed the Text-Line Segmentation Intrinsic Complexity Index (TLS-ICI), which measures complexity in ancient handwritten documents through intrinsic features like the HPP, VPP, HCB, and HCI. Afterward, a hybrid approach is proposed that selects the best alternative to solve automatic TLS, depending on the level of complexity of documents. The obtained results of this approach were compared to individual TLS methods through eight collections of ancient and contemporary documents, providing a better understanding of documents and task performance.

Section 3.2 consisted of presenting state-of-the-art evaluation methods that have been of interest because they do not require human references. These methods employ the statistical co-occurrence of terms (ROUGE-C), comparison of topics through term-weighting schemes and SVD (LSA), and divergence measures (SIMetrix). Notice that different text model representations were used, such as the bag of words, N-grams, Skip-grams, and LCS. In general, each method has been tested on DUC and TAC datasets, using manual and semi-automatic metrics as reference, such as $Retention_w$, Coverage, and Pyramids.

In Section 3.3, the most recent approximations based on the combination of metrics were presented. On the one hand, ROSE was presented as a method to combine the relevance of seven ROUGE's well-known metrics, using the CCA via the Monte Carlo method (Conroy & Dang, 2008). On the other hand, the linear regression method was proposed in (Ellouze et al., 2016) to combine 21 linguistic and content-based metrics. Both ROSE and the linear regression were proposed under the assumption that different metrics can capture relevant content to evaluate summaries and thereby improve the performance of automatic evaluation. Besides, ROSE and the linear regression are based on the evaluation with and without human references, respectively.

The last part of this chapter (Section 3.3.3) presents SECO-SEVA as an evaluation method based on the linear optimization of 31 content evaluation measures derived from ROUGE-C, LSA, and SIMetrix methods. These measures are focused on several evaluation criteria that allow users to use a wide range of features to evaluate summaries. However, some measures (*e.g.*, ROUGE-C-L) show more affinity than others because they obtained high relevance values. Based on this, the next chapter will present the proposed methodology that proposes a selection of content measures for the ETS, to improve automatic evaluation.

CHAPTER 4.

Proposed Methodology

Regarding the research question in the problem statement (see Section 1.1), selecting evaluation measures is essential because the most appropriate criteria should be determined to evaluate each summary. However, following a set of well-defined steps is necessary to perform such a selection. Therefore, this chapter outlines the proposed selection of content metrics under a methodology of six well-established steps. Such stages are presented in Figure 35.

The stages shown in Figure 35 were carried out sequentially because each is directly related to its predecessor. We also observe that this methodology provides feedback to the selection of evaluation metrics because of the following reasons:

1. Including new complexity indexes may improve the selection process.
2. Depending on the results obtained from Stage 6, we can adjust the parameters to improve the selection of measures.

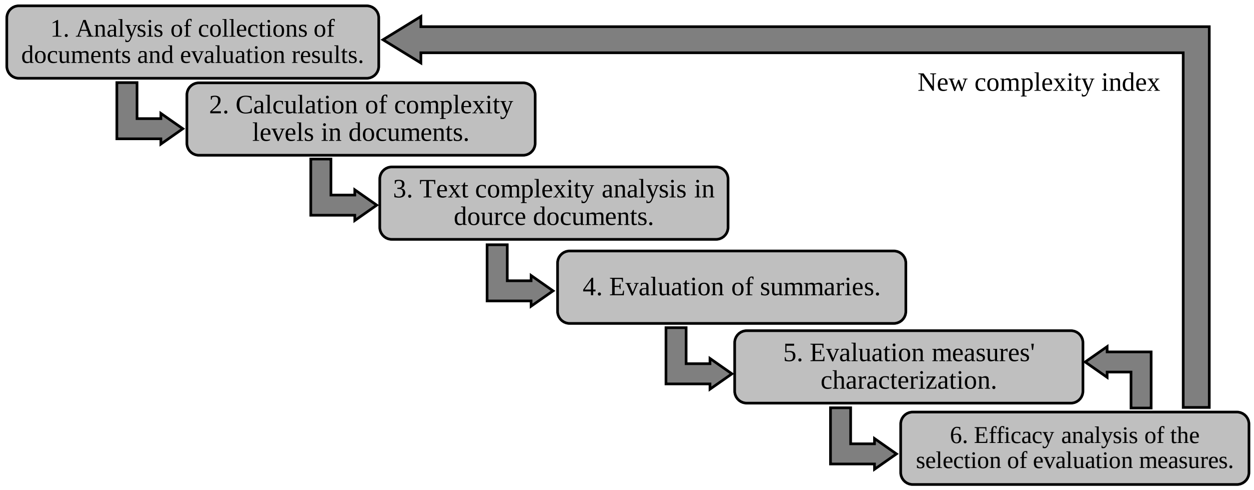


Figure 35. Stages of the proposed methodology.

With this methodology, we seek a feasible and unsupervised selection of evaluation measures that do not require human references. In this chapter, it is described how these stages were carried out.

4.1 Analysis of collections of documents and evaluation results

The analysis of collections of documents and the evaluation results is the first step of the proposed methodology. This step allows knowing the characteristics and requirements of documents to be processed. According to Mendoza et al. (2021), DUC01 and DUC02 have been distinguished as standard collections of documents to generate and evaluate summaries (Over, 2001; Over & Ligget, 2002). Furthermore, both collections provide documents that deal with a wide variety of topics such as natural disasters, bibliography information, technology, politics, and food. Based on this, DUC01 and DUC02 have been chosen to evaluate summaries.

Before evaluating summaries of both collections, it is necessary to identify the metrics and procedures carried out in the manual assessment. Table 6 shows the manual assessments and the number of evaluated summaries of each collection. While in DUC01, the manual evaluation was performed through the Weighted Retention measure ($Retention_w$), the Coverage measure was used in DUC02. Afterward, the results provided by the manual evaluation are used as a reference to measure the proximity between the proposed selection of metrics and human judgments. Once the manual assessment results are obtained, the content of summaries and source documents are identified. This way, source documents' content will determine suitable preprocessing procedures for calculating complexity levels.

Table 6. General description of DUC01 and DUC02 datasets.

Feature	DUC01	DUC02
# Source documents	309	567
# Collections	30	59
Manual assessment metric	$Retention_w$	Coverage
# Documents with manual assessment	1776	4107

According to the description of Table 6, DUC01 contains 309 news articles written in English grouped into 30 collections; each collection approximately contains 10 documents on average. These documents are based on news about natural disasters and bibliography information. For manual evaluation, some recall-based evaluation metrics were used to assess the summaries made by ATS systems. However, $Retention_w$ was used as a standard manual evaluation metric (Lin & Hovy, 2002). For single-document summarization task of 100 words, 12 ATS systems were manually evaluated, providing 1776 candidate summaries.

DUC02 contains 567 news articles written in English grouped into 59 collections; each collection has between 5 and 12 news articles about technology topics, food, politics, finances, and others. On DUC02, the manual evaluation was performed by Coverage metric. For single-document summarization of 100 words, 14 ATS systems were manually evaluated, providing 4107 candidate summaries.

Finally, it is worth mentioning that all documents of both have been translated into Spanish via Python Google Translate API.[19] This process has been done because we seek that the proposed selection of measures can be performed without language limitations. Therefore, it is expected that this approach can be language-independent.

4.2 Calculation of complexity levels in documents

In this stage, complexity levels in source documents are computed since the selection of evaluation metrics depends on each source document's complexity values. Therefore, all source documents

[19] Python library available at: https://pypi.org/project/googletrans.

of DUC01 and DUC02 are processed by the normalized versions of text complexity indexes seen in Section 2.11. In addition, we propose adapting traditional ROUGE-N metrics as text complexity indexes as follows:

4.2.1 ROUGE-N-based complexity indexes

As it has described in Section 2.6.3, ROUGE is an evaluation package whose underlying methods measure the similarity between the summary and human references. For instance, ROUGE-N measures the similarity of both documents through the overlap of n-grams, and resultant values are expressed in terms of Recall (R), Precision (P), and F-measure (F) (Lin, 2004). However, we can also use these measures as text complexity indexes by estimating the similarity between the source document (d_{src}) and the summary of the *Baseline:first*[20] (denoted as $S_{B:f}$), as shown in Equation 51, Equation 52, and Equation 53.

$$R_{ROUGE-N}(d_{src}, S_{B:f}) = \frac{\sum_{D \in \{d_{src}\}} \sum_{gram_n \in D} Count_{match}(gram_n)}{\sum_{D \in \{d_{src}\}} \sum_{gram_n \in D} Count(gram_n)}$$

Equation 51. ROUGE-N recall complexity index.

$$P_{ROUGE-N}(d_{src}, S_{B:f}) = \frac{\sum_{D \in \{d_{src}\}} \sum_{gram_n \in D} Count_{match}(gram_n)}{\sum_{D \in \{d_{src}\}} \sum_{gram_n \in D} Count(gram_n) + \sum_{gram_n \in S_{B:f}} Count_{unmatch}(gram_n)}$$

Equation 52. ROUGE-N precision complexity index.

$$F_{ROUGE-N}(d_{src}, S_{B:f}) = \frac{2 \times P_{ROUGE-N} \times R_{ROUGE-N}}{P_{ROUGE-N} + R_{ROUGE-N}}$$

Equation 53. ROUGE-N f-measure complexity index.

$R_{ROUGE-N}$ calculates the ratio of n-grams that co-occurs between d_{src} and $S_{B:f}$. The variable n represents the length of the n-gram, $Count(gram_n)$ is a function that counts the number of n-grams in d_{src}, and $Count_{match}(gram_n)$ counts the number of n-grams that match between d_{src} and $S_{B:f}$. The index $P_{ROUGE-N}$ measures the fraction of n-grams that $S_{B:f}$ and d_{src} have extracted in common, considering the number of unmatched n-grams of both documents ($Count_{unmatch}(gram_n)$). Finally, $F_{ROUGE-N}$ calculates the harmonic mean between $R_{ROUGE-N}$

[20] Baseline:first is a heuristic that extracts the first sentences from a source document to generate an extractive summary of a predefined length of words.

and $P_{ROUGE-N}$. The primary purpose of adapting ROUGE-N metrics is to measure the degree to which source documents convey similar information compared to their summaries constructed from the first sentences.

4.2.2 Average Complexity Index

Besides complexity indexes described in Section 2.11, we propose the use of an arithmetic average of all indexes to estimate the overall complexity of each source document in the DUC01 and DUC02 datasets. In view of this, the Average Complexity Index (ACI) is proposed and displayed in Equation 54, where $Index_i(d_{src})$ represents the complexity value associated with the i^{th} index, which also is calculated from the source document d_{src}; n stands for the number of text complexity indexes included.

$$ACI(d_{src}) = \sum_{i=0}^{n} Index_i(d_{src})$$

Equation 54. Average Complexity Index.

4.3 Text complexity analysis in source documents

Once each source document's complexity level has been calculated, the distributions of complexities are analyzed in this step. Therefore, the source documents of DUC01 and DUC02 are sorted from lowest to highest according to the value of each normalized index ($Index_i$) and the ACI. In other words, it provides orders of complexity for each collection of documents.

4.4 Evaluation of summaries

The evaluation of text summaries is the fourth step of the proposed methodology on which a selection of evaluation metrics is performed. For this, we have considered the selection of metrics previously proposed by Rojas-Simón et al. (2021). Thus, both automatic and manual summaries are evaluated from 31 metrics derived from ROUGE-C (He et al., 2008), LSA (Steinberger & Ježek, 2009), and SIMetrix (Louis & Nenkova, 2013).

Concerning the list of metrics shown in Table 3, ROUGE-C can evaluate under n-grams of lengths from 1 to 5, generating the following metrics: ROUGE-C-1, 2, 3, 4, and 5. Moreover, this method uses Longest Common Subsequences (LCS) and Skip-Unigrams (SU), obtaining ROUGE-

C-L and ROUGE-C-SU4, respectively. For the LSA-based evaluation, local and global term-weighting formulas seen in Section 2.6.1.3 have been used, resulting in 16 metrics. The letters shown between parentheses indicate a specific term-weighting combination. Finally, it has been selected smoothed ($JS–Smth_n$) and unsmoothed (JS_n) versions of the JS divergence from SIMetrix, where n stands for the length of n-gram. Despite 31 ways to evaluate summaries being provided, we can add more metrics to the proposed selection.

4.5 Evaluation measures' characterization

The characterization of evaluation measures is a step of the proposed methodology that assigns a level of complexity to each measure. Subsequently, the documents' complexity levels calculated from Step 2 (see Section 4.2) are used as a reference to select the most appropriate evaluation metric. In general, complexity of evaluation measures can be determined according to the following ways:

4.5.1 *A priori*

This characterization is the most common-sense approach because the complexity of each measure may depend on its proximity toward human judgments. Thus, this proximity is measured via correlation coefficients explained in Section 2.8. In this way, correlation values are associated with an order of complexity since while a measure shows a higher correlation, the more complex it is. Based on this criterion, the average among Pearson, Spearman, and Kendall correlations is used to establish specialization ranges for each measure.

4.5.2 Characterization via Genetic Algorithm (GA)

As mentioned in Section 2.12, the Genetic Algorithm (GA) is one of the most well-known and used evolutionary optimization techniques. Its use in the ATS and ETS has proven to be a suitable alternative for creating and evaluating summaries. Therefore, it is employed to estimate the complexity of each measure. This section briefly explains the proposed GA's encoding and operators.

4.5.2.1 Encoding representation

The complexity of each measure is represented in a vector of n segments. Each j^{th} segment (w_j) shows the complexity of one evaluation measure in a sub-vector of n_s genes. Therefore, each sub-vector represents in binary values a decimal value in a range of 0.0 (lowest complexity) to 1.0 (highest complexity), as shown in Equation 55.

$$X_i(g) = [S_1, S_2, \dots, S_n], w_j = \left[X_{i,1}(g), X_{i,2}(g), \dots, X_{i,n_s}(g)\right], X_{i,k}(g) \in \{0,1\}$$

Equation 55. Proposed GA's encoding.

4.5.2.2 Proposed operators

This section describes the proposed GA operators to calculate the complexity of each evaluation measure.

4.5.2.2.1 Initial population

The most common strategy to initialize a population of N_{pop} chromosomes (when $g = 0$) consists of generating random binary values for each k^{th} gene from each i^{th} chromosome $\left(X_{i,k}(0) = Random[0,1]\right)$. Therefore, each gene of each s_j segment has the probability of selecting a binary value between 1 and 0. However, the chain of genes that represent the s_j segment depicts decimal values in a range of [0.0–1.0].

4.5.2.2.2 Fitness function

The proposed fitness function is based on the formula shown in Equation 56, where a_{lk} represents the evaluation score of the measure k for each l^{th} candidate summary. The k^{th} measure belongs to the set of n measures, and it is selected if its complexity value ($C(a_k)$) is greater than or equal to the average complexity index of the source document ($ACI(d_{src})$).

$$f\left(X_i(g)\right) = \frac{1}{R}\sum_{j=1}^{R} r_j(a_{lk}, b_l), a_k \in \{a_1, \dots, a_n\}, a_k \; is \; selected \; if \; C(a_k) \geq ACI(d_{src})$$

Equation 56. Proposed fitness function.

Notice that such selection is performed for each l^{th} candidate summary. Therefore, we seek to maximize the correlation of such selection concerning $Retention_w$ and Coverage, considering R correlation coefficients, where r_j is the correlation value obtained from the j^{th} coefficient.

4.5.2.2.3 Selection

Selection operators use the fitness value of chromosomes to choose parents with better characteristics (genes). According to evolutionary principles, the probability of better solutions is increased if the genetic exchange is performed between two chromosomes with good fitness values. For the proposed GA, two selection operators were used. The first one is the *elitism selection* that guarantees the n_{elite} best chromosomes of each generation are chosen. The second operator is the *tournament selection*. This operator generates $N_{pop} - n_{elite}$ samples of K_{Tourn} chromosomes (randomly chosen) from the population to select the best ones of each sample.

4.5.2.2.4 Crossover

As mentioned in Section 2.12.1, crossover operators perform the genetic exchange between two chromosomes to obtain better offspring. For the proposed GA, the *uniform crossover* operator was used. Figure 36 illustrates a practical example of how this operator works, where $X_{p_1}(g)$ and $X_{p_2}(g)$ are two selected parents from the tournament selection operator. Each gene has a probability of being selected as a crossover point, depending on the probability value P_c specified as a parameter.

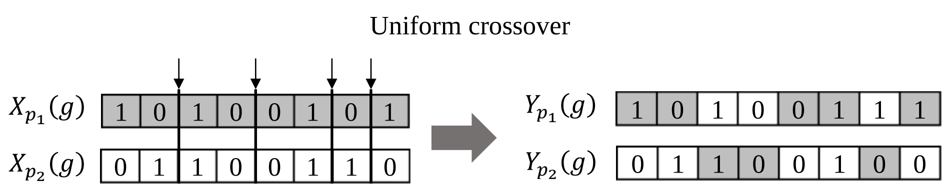


Figure 36. Uniform crossover.

4.5.2.2.5 Mutation

As explained in Section 2.12.1, the mutation step selects a set of genes from a population of $Y_i(g)$ chromosomes, modifying its values to generate a new population denoted as $Z_i(g)$. For the proposed GA, the *flipping operator* was used, which inverts the value of the m gene in the chromosome $Y_i(g)$, according to Equation 57:

$$Z_{i,m}(g) = \begin{cases} Random[0,1], & if\ 0 < p_m \le P_m \\ Y_{i,m}(g), & otherwise \end{cases}$$

Equation 57. Flipping operator.

where P_m represents the mutation probability in a range of $[0.0–100.0]$. Therefore, if the random value p_m is in the range $(0–P_m]$, then the following random function is applied: $Random[0,1]$. Otherwise, the gene m is not modified ($Y_{i,m}(g)$).

In addition to the flipping operator, the *cataclysmic mutation* operator is also included, preserving the diversity of chromosomes through the restart procedure (Eshelman, 1991). Therefore, this operator is applied in the last generation of the GA, selecting the best chromosome from such generation. Afterward, the remaining chromosomes are randomly generated to initialize a new population of N_{pop} chromosomes and restart the GA again (Ledeneva & García-Hernández, 2017; Rathee & Ratnoo, 2020). Considering this procedure, the number of restarts ($\#Restarts$) is used as a parameter in the proposed GA.

4.5.3 Characterization via Multilayer Perceptron (MLP)

The last characterization considered in this thesis is performed through a Multilayer Perceptron (MLP), which is a particular Feedforward Neural Network (FNN) commonly used to solve classification and regression problems. Based on the idea that MLP can solve the above-mentioned types of problems, this section describes how MLP was used to characterize each evaluation measure.

4.5.3.1 Classification

In literature, classification is seen as a particular case of regression whose outputs are categorical data. Based on this, we address the selection of measures as a classification problem, where each summary is assigned to a particular evaluation measure. Therefore, the proposed MLP should determine the most appropriate measure in the output layer according to the complexity of each source document (inputs). Figure 37 illustrates the overall scheme of this characterization. As observed, the 13 complexity values of each source document are introduced as inputs to the MLP.

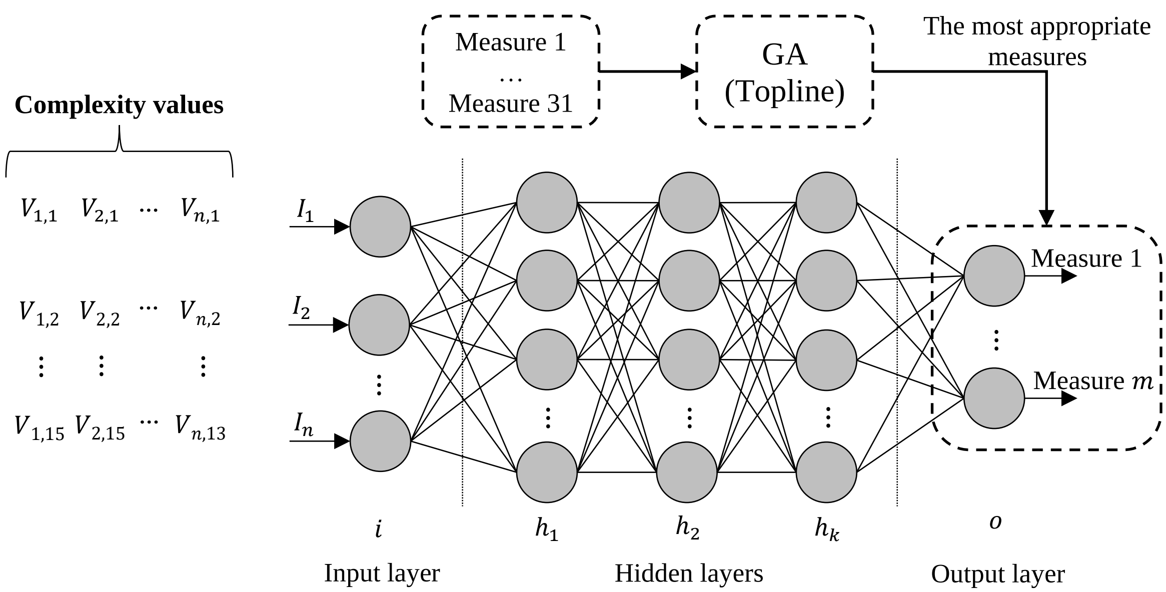


Figure 37. Characterization of measures via classification.

Afterward, the MLP performs the optimization of weights to obtain meaningful patterns that help to characterize the evaluation of each summary, assigning them a label class that represents a particular measure. However, it is necessary to know *a priori* which are the most suitable evaluation measures for each source document. To achieve an affordable set of measures, we employ the GA to obtain an approximation to the best selection of measures.

The encoding representation for this GA is called *multi-base*, whose genes are integer numbers representing each measure's identifier. Equation 58 formally describes this representation in an array of integer numbers, where $X_i(g)'$ stands for the i^{th} chromosome in the generation g. Each k^{th} gene is associated with a particular evaluation measure, and n represents the length of each chromosome, which is the number of summaries to evaluate. Therefore, the chromosomes in DUC01 and DUC02 consist of 1776 and 4107 genes, respectively.

$$X_i(g)' = \left[X_{i,1}(g)', X_{i,2}(g)', \dots, X_{i,n}(g)'\right], X_{i,k}(g)' \in \{0, \dots, 30\}$$

Equation 58. Encoding representation for characterization.

Considering the before-explained encoding representation, we calculate the fitness of each chromosome according to Equation 59. In general, the proposed function calculates the average of R correlation coefficients between the scores obtained from the selection of the k^{th} measure for each l^{th} summary and their corresponding scores obtained from human judgments. For the proposed experimentation (see Chapter 5), the Pearson, Spearman, and Kendall coefficients were included.

$$f\left(X_i(g)\right)' = \frac{1}{R}\sum_{j=1}^{R} r_j(a_{lk}, b_l), a_k \in \{a_0, \dots, a_{30}\}$$

Equation 59. Fitness function for characterization.

Finally, the GA operators for this GA and their parameters are the following:

- Selection: Elitism (n_{elite}) and Tournament (K_{Tourn}).
- Crossover: Uniform (P_c).
- Mutation: Multi-base flipping operator (p_m) and Cataclysmic ($\#Restarts$).

Moreover, it is worth mentioning that the best chromosome of this GA is used as a reference to obtain the Topline (upper bounds) because it represents an approximation to an ideal selection of evaluation of measures for each source document. Thus, the progress of the proposed selection of measures can also be calculated.

4.5.3.2 Regression

Regression is a supervised-learning technique that seeks to predict a continuous output variable based on the input features of studied objects. It is widely used in biological, environmental, economic, and social sciences to predict the phenomena of different processes. For neural networks, regression is used to predict the behavior of output variables finding linear and non-linear associations. Based on this characteristic, this characterization proposes using the MLP to predict the evaluation score of each summary according to the 13 complexity values calculated from each corresponding source document.

Figure 38 illustrates the overall scheme of this characterization, where complexity values of each source document are introduced as inputs to the MLP. Then, the MLP performs the adjustment of weights to generate an output signal, whose value represents the evaluation of each summary.

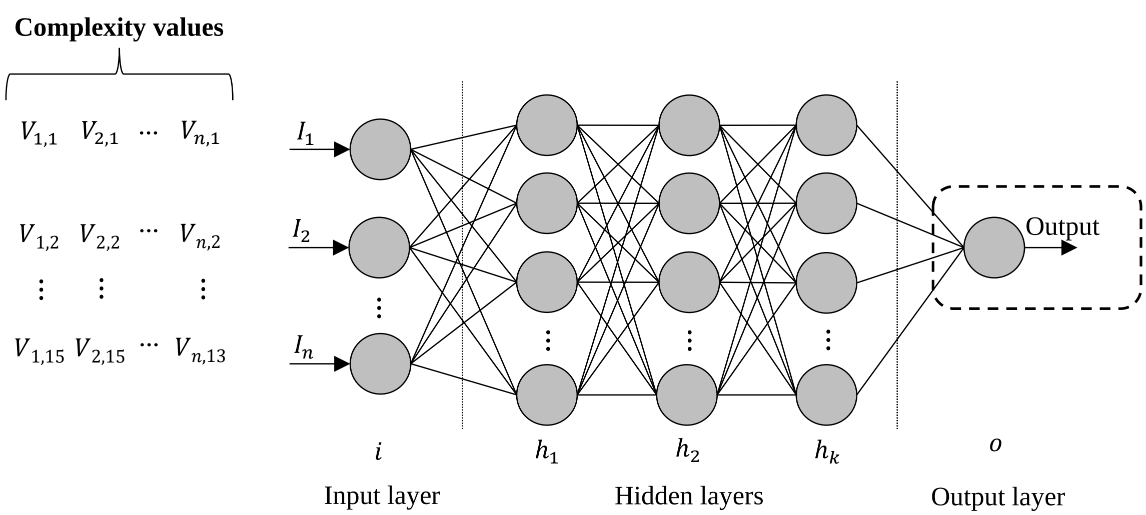


Figure 38. Characterization via regression.

4.6 Efficacy analysis of the selection of evaluation measures

In the last stage, the analysis of linear optimization is an important stage that measures the correlation of the best combination of metrics concerning human judgments. Such analysis is reflected in three different aspects. The first one measures the correlation of linear optimization of metrics on micro and macro evaluation levels through tuning GA parameters. Tuning GA parameters intends to improve the optimization of the GA, modifying the number of generations, size of the population, size of the tournament, probability of crossover and mutation, and the number of restarts. The second one is related to the relevance analysis of the best combination of metrics provided by the GA. The resultant relevance values indicate if a metric is useful and important.

Finally, the third aspect is based on the comparison of the best combination of metrics with the best individual evaluation metrics, in terms of the Pearson, Spearman, and Kendall correlation coefficients. This comparison allows comparing the effectiveness of the best combination of metrics on micro and macro evaluation levels on DUC01 and DUC02. Moreover, each comparison based on different correlation coefficients and evaluation levels was considered for providing a unified ranking of evaluation metrics.

4.7 Summary of the chapter

In this chapter, the proposed methodology was presented, which is based on six stages to perform the selection of evaluation methods. The first stage of this methodology is based on dataset analysis. Within this stage, DUC01 and DUC02 datasets were used to evaluate the performance of evaluation measures.

Subsequently, the selection of automatic metrics is the second stage of the proposed methodology that determines the set of metrics to combine. In this stage, different text representation models and term-weighting schemes were used to generate 31 evaluation metrics derived from ROUGE-C, LSA, and SIMetrix methods. The evaluation of summaries is the third stage, where all summaries (micro evaluation) and ATS systems (macro evaluation) were evaluated from each evaluation metric. Before evaluating summaries, all documents were preprocessed through the elimination of stopwords and Porter's stemming.

Finally, the correlation analysis of the linear optimization of metrics represents the last stage of the proposed methodology. Such correlation analysis is addressed in three different aspects. The first one is based on tuning GA parameters. The second one is related to the relevance analysis of the best combination of metrics provided by the GA. The third aspect is based on the comparison of the best combination of metrics with the best individual evaluation metrics, in terms of the

Pearson, Spearman, and Kendall correlation coefficients.

CHAPTER 5.

Experiments and Results

The results of applying the proposed methodology are divided as follows: First, we show an analysis of text complexity distributions in the DUC01 and DUC02 datasets. Afterward, we present the characterization of evaluation metrics, whose obtained values indicate decision thresholds in which each evaluation metric should be selected. Finally, the correlation results between the proposed selection of metrics and human judgments are shown, and we compare the performance of such selection concerning state-of-the-art automatic metrics.

5.1 Analysis of text complexity distributions

Analyzing complexity in source documents is crucial since we can examine: (i) the distribution values of state-of-the-art complexity indexes and (ii) their suitability to the proposed selection of metrics. The box diagram shown in Figure 39 displays text complexity distributions on documents of the DUC01 and DUC02 datasets. Notice that all complexity values vary from 0.0 to 1.0 because these indexes have been previously normalized, providing them with the same degree of relevance.

The TTR index shows a distribution of values near the middle area of the diagram, which means most documents in DUC01 and DUC02 present moderate diversity of words. However, there are documents whose complexity values are higher than 0.6. On the other hand, the index RSW shows similar complexity values in both datasets because their mean values are closer to 0.6. These results indicate approximately 60% of the tokens of each source document are stop words. Concerning the morphology of words, the RIW index displays higher and more compacted distributions of complexities, whose obtained scores are near the mean of 0.7. Therefore, we can infer that, on average, 70% of tokens represent words acquired any inflection by adding prefixes and suffixes.

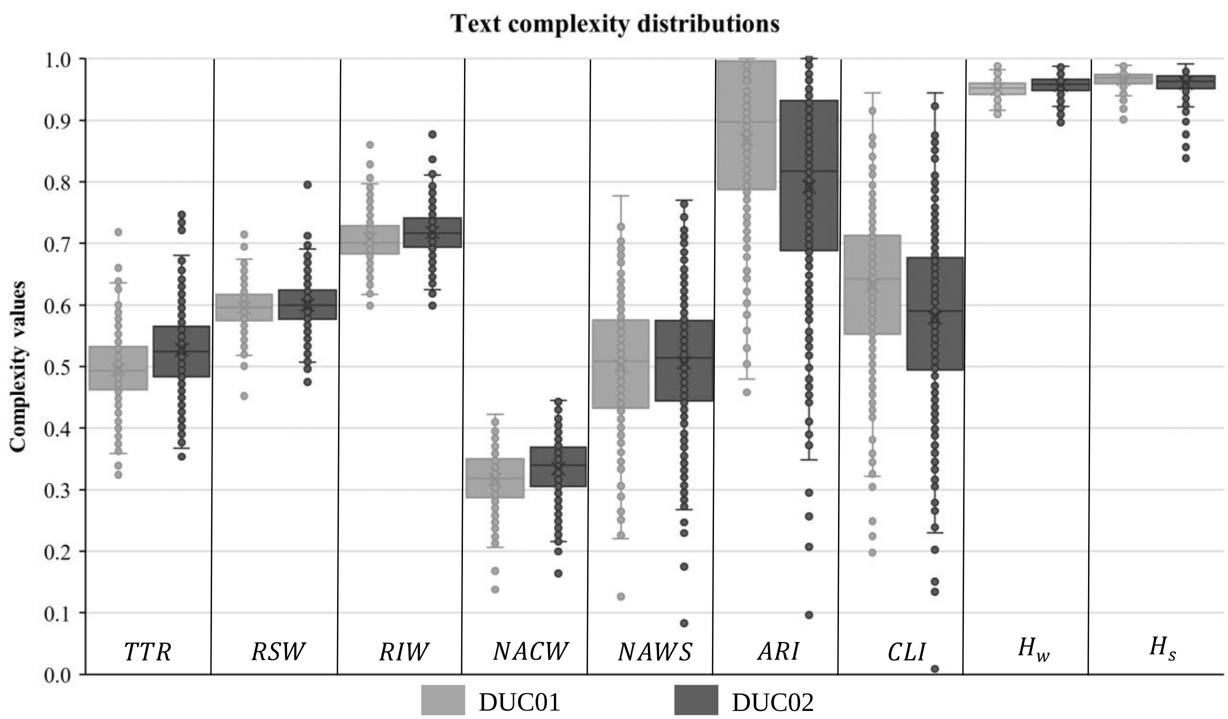


Figure 39. Distribution of complexities in DUC01 and DUC02 using state-of-the-art complexity indexes.

Besides linguistic diversity and morphology of words, the average length of words/sentences has also been analyzed through NACW and NAWS indexes. According to the NACW index, documents of the DUC01 and DUC02 datasets show complexity values near 0.3, indicating a low uniformity of word lengths. In other words, there is a significant difference between the average of characters per word and the longest word of each source document. By contrast, the NAWS index shows complexity scores from 0.2 to 0.8. These values indicate that source documents have variable sentence lengths.

According to the ARI and CLI, most readability values range from 0.3 to 1.0 (except for outliers) in both collections of source documents. Compared to other complexity indexes, readability indexes show the broadest distributions of values, so we can easily differentiate complexity values throughout collections. In the case of ARI, most readability scores are in the range of 0.7 to 1.0, and most CLI values range from 0.5 to 0.7. Moreover, we observe that ARI has provided the highest complexity value (1.0) to documents of both collections, which means some documents are challenging to read and, therefore, these values determine that the most complex evaluation metrics evaluate their corresponding summaries.

On the other hand, the entropy-based indexes (H_w and H_s) present the most compacted distributions in the comparison. Obtained values generally range from 0.9, indicating a wide diversity of words and variable sentence lengths. In other words, we assume that most words found in each source document (excluding stopwords) are used only once. On the other hand, H_s indicates that most sentences in each document show significantly varying lengths. This situation is expected since many words are used only once in the document, and every sentence tends to have different sizes.

In addition to the text complexity distributions shown in the previous diagram, we have experimented with the indexes described in Section 4.2. The obtained distributions are shown in Figure 40. As we observe, the ROUGE-N-based indexes show the broadest distributions of values. Therefore, we note that these indexes can clearly differentiate the complexity of source documents of both collections. On the other hand, we observe that their distributions of values are similar. This is because both $R_{ROUGE-N}$ and $P_{ROUGE-N}$ consider the number of n-grams that co-occurs between the $S_{B:f}$ and the d_{src}. However, $P_{ROUGE-N}$ adds a penalization using the number of unmatched n-grams between both documents. Moreover, $F_{ROUGE-N}$ represents the harmonic mean of distributions between $R_{ROUGE-N}$ and $P_{ROUGE-N}$.

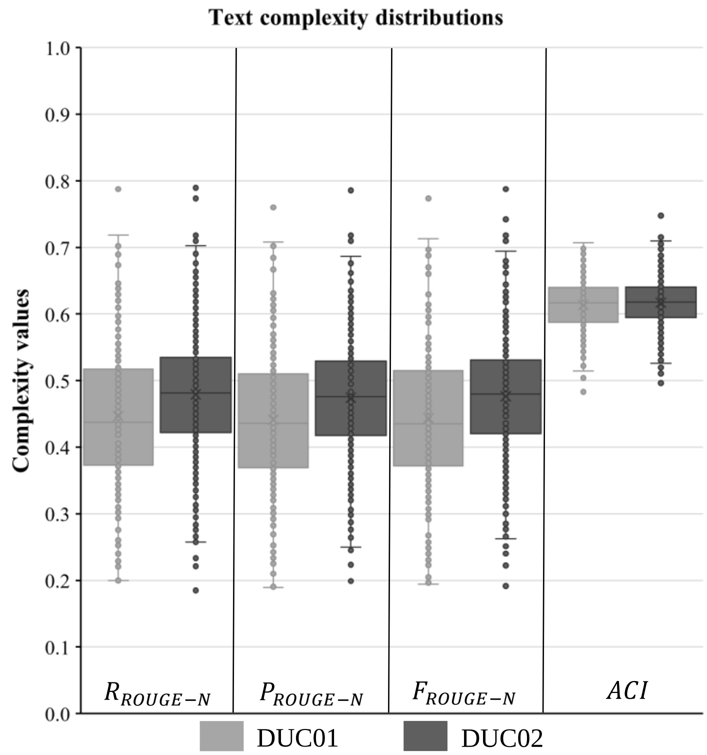


Figure 40. Distribution of complexities in DUC01 and DUC02 using ROUGE-N-based indexes and the ACI.

Finally, we have experimented with the ACI, which represents an average of 12 text complexity indexes. Unlike ROUGE-N-based and readability indexes, the ACI shows a more compacted distribution of values. This is because other indexes provide more compacted distributions, such as the H_w, H_s, RSW, and RIW (see Figure 39). In view of this, we assume that compacted distributions imply that each source document would not be clearly differentiated to the selection of metrics. Thus, the characterization of content-based metrics needs to be fine-tuned to differentiate the evaluation of each summary.

5.2 Characterization of content metrics

According to the fifth step of the proposed methodology (see Section 4.5), the characterization of content metrics consists of assigning each a complexity value according to its proximity to human judgments. Nevertheless, this proximity is usually measured with different correlations, such as Pearson, Spearman, and Kendall measures. Therefore, to indicate the overall complexity of content metrics, we have calculated the harmonic mean among Pearson, Spearman, and Kendall correlation results of each 12 best state-of-the-art metrics previously reported by Rojas-Simón et al. (2021). The obtained results are shown in Figure 41.

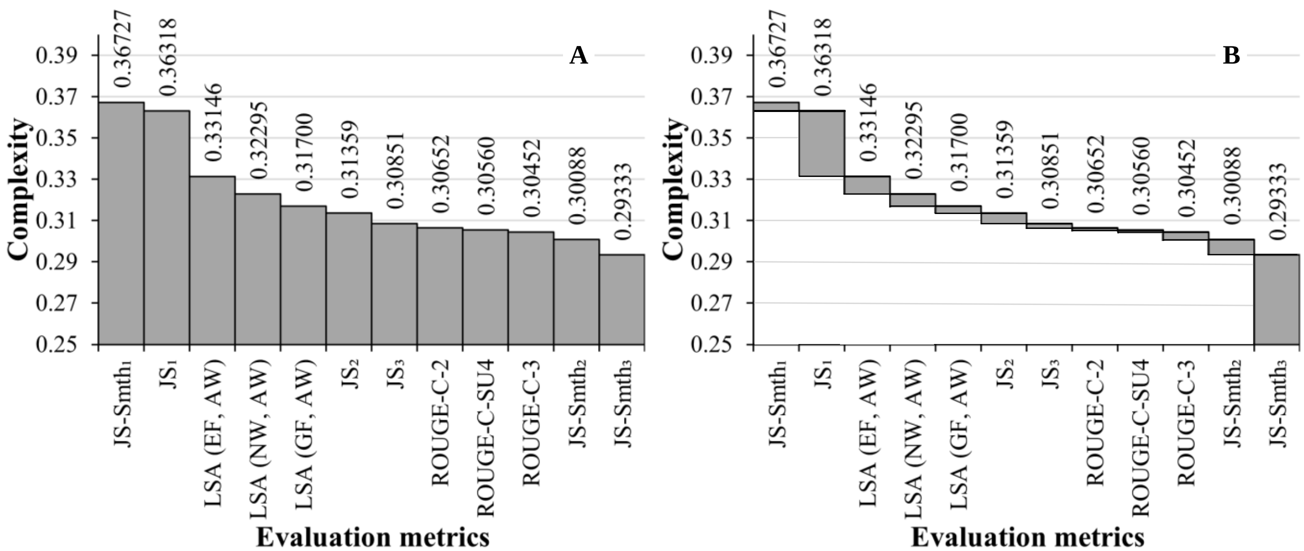


Figure 41. Complexity values of state-of-the-art evaluation metrics on the DUC01 dataset.

On the left side of Figure 41 (A), we observe a ranking of complexities of state-of-the-art metrics on the DUC01 dataset. First, we observe that $JS\text{–}Smth_1$ obtained the highest complexity value (0.36727) since it obtained the best correlation results (Pearson: 0.43313, Spearman: 0.41682, and Kendall: 0.28897). Afterward, this metric is followed by JS_1, which obtained the second-highest complexity (0.36318) because it obtained high results on individual correlations (Pearson: 0.43015, Spearman: 0.41325, and Kendall: 0.28443). Next, the LSA-based metrics obtained complexities equal to 0.33146, 0.32295, and 0.31700, respectively. However, they show an evident gap concerning $JS\text{–}Smth_1$ and JS_1. Finally, the remaining metrics show complexity values near 0.30.

Based on the ranking of metrics shown in Figure 41A, we note that an overall complexity value has characterized each metric. While better metrics show higher complexity values, worse metrics show the opposite. Therefore, we assume that each is specialized in a range of complexity. Such ranges are necessary to indicate what metric should be selected depending on the complexity value of each source document to evaluate summaries.

Figure 8B shows the complexity ranges used to choose each metric, creating a Hybrid Evaluation Method (HEM). From these ranges, it is observed that both JS_1 and $JS\text{–}Smth_3$ can cover other metrics because both metrics are specialized in broader ranges. Moreover, we notice although $JS\text{–}Smth_1$ is showing the highest complexity value, it is limited in a short gap concerning other metrics. To address this situation, we expanded the selection of this metric to documents of higher complexity.

5.3 Performance of the proposed selection of content metrics

The tables presented in this section show the correlation results between the automated selection of metrics and state-of-the-art metrics on the DUC01 and DUC02 datasets. Table 7 displays that the selection of metrics based on RSW and PRW indexes correlate better with human judgments than the best individual metric of the state-of-the-art ($JS - Smth_1$) in Pearson, Spearman, and Kendall coefficients. This is because of the influence and presence of stopwords and inflected words in each evaluated summary.

Table 7. Correlation results of the selection of metrics and state-of-the-art methods on DUC01.

	Index/Metric	Pearson	Spearman	Kendall
Selection of metrics	TTR	0.42833	**0.41891**	**0.29181**
	RSW	**0.43313**	**0.41682**	**0.28897**
	PRW	**0.43327**	**0.41710**	**0.28903**
	ACW	0.25082	0.30117	0.21165
	AWS	0.39628	0.39208	0.27120
	H_w	0.43313	0.41682	0.28897
	H_s	0.43313	0.41682	0.28897
	ARI	0.43313	0.41682	0.28897
	CLI	**0.43852**	**0.42061**	**0.29143**
	$ROUGE_{Recall}$	0.37847	**0.43539**	**0.29748**
	$ROUGE_{Precision}$	0.39998	**0.43818**	**0.30088**
	$ROUGE_{F-measure}$	0.39573	**0.43576**	**0.29710**
State-of-the-art	$JS - Smth_1$	**0.43313**	**0.41682**	**0.28897**
	JS_1	0.43015	0.41325	0.28443
	LSA-3	0.39530	0.37296	0.26042
	LSA-1	0.38541	0.36538	0.25267
	LSA-2	0.37830	0.35798	0.24832

Note: Best correlation results are highlighted in bold.

Moreover, measuring complexity with the CLI was helpful in selecting metrics because it obtained the best results in Pearson, Spearman, and Kendall coefficients. On the other hand, selecting metrics under other complexity indexes such as TTR, $ROUGE_{Recall}$, $ROUGE_{Precision}$, and $ROUGE_{F-measure}$ obtained better results than individual evaluation metrics on Spearman and Kendall coefficients.

According to the correlation results in Table 8, the ROUGE's measurements as complexity indexes showed the best performance under the Pearson, Spearman, and Kendall correlations, outperforming the best individual metric ($JS - Smth_1$). Such results suggest that the overlap of n-grams between source documents and candidate summaries is another helpful indicator for estimating complexity in summaries and selecting the most appropriate evaluation metrics. Furthermore, the selection under other indexes such as the TTR and AWS also showed improvements in the Spearman and Kendall correlations to a lesser degree.

Table 8. Correlation results of the selection of metrics and state-of-the-art methods on DUC02.

	Index/Metric	Pearson	Spearman	Kendall
Selection of metrics	TTR	0.34640	**0.36362**	**0.25538**
	RSW	0.38979	0.33525	**0.26201**
	PRW	0.38979	0.35835	**0.25241**
	ACW	0.32639	0.32856	0.22534
	AWS	0.31138	**0.35974**	**0.25508**
	H_w	0.38979	0.35835	0.25094
	H_s	0.38979	0.35835	0.25094
	ARI	0.36983	0.34887	0.24513
	CLI	0.31838	0.32353	0.22529
	$ROUGE_{Recall}$	**0.40427**	**0.41027**	**0.28432**
	$ROUGE_{Precision}$	**0.45602**	**0.42903**	**0.29919**
	$ROUGE_{F-measure}$	**0.40945**	**0.41650**	**0.28951**
State-of-the-art	$JS - Smth_1$	**0.38979**	**0.35835**	**0.25094**
	JS_1	0.37907	0.34807	0.24431
	LSA-3	0.36201	0.33154	0.23254
	ROUGE-C-3	0.38789	0.32134	0.22543
	LSA-1	0.35916	0.32966	0.23130

Alternatively, the indexes RSW and PRW showed improvements in the Kendall correlation. However, although the selection under other indexes showed lower performance, some of them show an evident approximation to the best individual metric. This observation is important to consider because the combination of complexity indexes may be useful way to measure the overall complexity of summaries.

CHAPTER 6.

Conclusions and future works

The ETS has been an essential task immersed in the ATS that seeks to measure the performance of proposed methods/systems under different criteria (extrinsic and intrinsic). For intrinsic evaluation, most of the proposed methods compare the content between the candidate summary and human references. However, human references are summaries that are not always available, such that their creation is expensive and time-consuming. Due to this, ROUGE-C, LSA, and SIMetrix have been proposed as evaluation methods that do not require human references.

Previous works suggest the linear optimization of their metrics might improve automatic evaluation, but it involves adjusting internal parameters, indicating an implicit degree of complexity. In this thesis, it is proposed the analysis of complexity in text summaries and evaluation metrics to select the most appropriate metrics for each candidate summary.

The proposed methodology of six steps (described in Chapter 4) is guided as a linear and incremental process because of two principal reasons: (i) it requires that each step depends on its predecessor(s), and (ii) it allows for including more complexity indexes. In this sense, while more indexes are included, there will be more criteria to perform the selection of content metrics. According to the correlation results shown in Tables 2 and 3, the selection of content metrics under complexity indexes improves the performance of automatic evaluation without human references, outperforming state-of-the-art individual evaluation metrics. As observed, the selection with some indexes such as TTR, $ROUGE_{Recall}$, $ROUGE_{Precision}$, and $ROUGE_{F-measure}$ show better performance than the rest of complexity measures. However, it is important to consider that a linear combination of complexity measures will obtain a better performance in selecting metrics.

6.2 Publications

Till the moment, the following manuscript and book have been published as a result of this work:

- **[Article]** Rojas-Simón, J., Ledeneva, Y., & García-Hernández, R. A. (2021). Evaluation of text summaries without human references based on the linear optimization of content metrics using a genetic algorithm. Expert Systems with Applications, 167, 113827. https://doi.org/10.1016/j.eswa.2020.113827
- **[Book]** Rojas-Simon, J., Ledeneva, Y., & Garcia-Hernandez, R. A. (2022). Evaluation of Text Summaries Based on Linear Optimization of Content Metrics (Vol. 1048). Springer International Publishing. https://doi.org/10.1007/978-3-031-07214-7
- **[Article]** Rojas-Simon, J., Ledeneva, Y., & García-Hernández, R. A. (2023). La evaluación de resúmenes biomédicos del COVID-19 sin referencias humanas.
- **[Article]** Carriola-Careaga, A., Rojas-Simon, J., Ledeneva, Y., & García-Hernández, R. A. (2023). Performance of Evaluation Methods without Human References for Multi-document Text Summarization. **Manuscript in submission stage for *Computer & speech language* journal.**

Moreover, the following manuscripts are in submission and review process:

- **[Article]** Rojas-Simon, J., Ledeneva, Y., & García-Hernández, R. A. (2022). Evolutionary Linear Optimization of Content Metrics for Multi-document Summarization Evaluation. **Manuscript in submission stage for *Knowledge-based systems* journal.**

- **[Article]** Rojas-Simon, J., Ledeneva, Y., & García-Hernández, R. A. (2022). Selection of Content Metrics for Evaluation of Text Summaries via Text Complexity Indexes. **Manuscript in submission stage for *Frontiers in Artificial Intelligence* journal.**

References

Alaei, A., Pal, U., & Nagabhushan, P. (2012). Dataset and ground truth for handwritten text in four different scripts. *International Journal of Pattern Recognition and Artificial Intelligence*, *26*(4), 25. https://doi.org/10.1142/S0218001412530011

Amigó, E. (2006). Sintesis De Informacion: Desarrollo Y Evaluacion De Un Modelo Interactivo. In *Spanish Society for Natural Language Processing* (Vol. 7). Universidad Nacional de Educación a Distancia (UNED).

Araujo, L., & Cervigón, C. (2009). *Algoritmos Evolutivos: Un Enfoque Práctico* (Second Edi). RA-MA.

Arivazhagan, M., Srinivasan, H., & Srihari, S. (2007). A statistical approach to line segmentation in handwritten documents. In X. Lin & B. A. Yanikoglu (Eds.), *Proc. SPIE 6500, Document Recognition and Retrieval XIV* (p. 65000T). https://doi.org/10.1117/12.704538

Arvanitopoulos, N., & Susstrunk, S. (2014). Seam Carving for Text Line Extraction on Color and Grayscale Historical Manuscripts. *2014 14th International Conference on Frontiers in Handwriting Recognition*, 726–731. https://doi.org/10.1109/ICFHR.2014.127

Bentz, C., Ruzsics, T., Koplenig, A., & Samardži, T. (2016). A Comparison Between Morphological Complexity Measures: Typological Data vs. Language Corpora. *Workshop on Computational Linguistics for Linguistic Complexity, Coling*, 142–153. https://www.aclweb.org/anthology/W16-4117/

Cabrera-Diego, L. A., & Torres-Moreno, J.-M. (2018). SummTriver: A new trivergent model to evaluate summaries automatically without human references. *Data & Knowledge Engineering*, *113*, 184–197. https://doi.org/10.1016/j.datak.2017.09.001

Cabrera-Diego, L. A., Torres-Moreno, J.-M., & Durette, D. (2016). Evaluating Multiple Summaries Without Human Models: A First Experiment with a Trivergent Model. *Lecture Notes in Computer Science*, *7934*(1), 91–101. https://doi.org/10.1016/S1003-6326(11)60841-8

CCSSO. (2010). *Supplemental Information for Appendix A of the Common Core State Standards for English Language Arts and Literacy: New Research on Text Complexity*. www.achievethecore.org/text-complexity.

Cha, S. (2007). Comprehensive Survey on Distance / Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, *1*(4), 300–307. https://doi.org/10.1007/s00167-009-0884-z

Chai, Y., Tao, S., Mao, W., Zhang, K., & Zhu, Z. (2018). Online incipient fault diagnosis based on Kullback-Leibler divergence and recursive principle component analysis. *The Canadian*

Journal of Chemical Engineering, *96*(2), 426–433. https://doi.org/10.1002/cjce.22962

Cheng, W., Greaves, C., & Warren, M. (2007). From n-gram to skipgram to concgram. *International Journal of Corpus Linguistics*, *11*(4), 411–433. https://doi.org/10.1075/ijcl.11.4.04che

Chisholm, E., & Kolda, T. G. (1999). New Term Weighting Formulas for the Vector Space Method in Information Retrieval. In *Computer Science and Mathematics Division*. https://doi.org/10.2172/5698

Coleman, M., & Liau, T. L. (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, *60*(2), 283–284. https://doi.org/10.1037/h0076540

Conroy, J. M., & Dang, H. T. (2008). Mind the Gap: Dangers of Divorcing Evaluations of Summary Content from Linguistic Quality. *Proceedings of the 22nd International Conference on Computational Linguistics*, *1*, 145–152.

Conroy, J. M., Schlesinger, J. D., Rankel, P. A., & O'Leary, D. P. (2010). Guiding CLASSY toward more responsive summaries. *Proceedings of the TAC 2010 Workshop*, 1–12. https://tac.nist.gov/publications/2010/participant.papers/CLASSY.proceedings.pdf

Cover, T. M., & Thomas, J. A. (2012). *Elements of Information Theory* (2nd ed.). John Wiley & Sons, Inc.

da Cunha, I. (2008). *Hacia un modelo lingüístico de resumen automático de artículos médicos en español* [Universitat Pompeu Fabra]. https://www.tesisenred.net/bitstream/handle/10803/7508/tidcf.pdf.pdf?sequence=1&isAllowed=y

Dang, H. T. (2010). Overview of the 2010 Text Analysis Conference. *Proceedings of the Text Analysis Conference*, 1–29. https://tac.nist.gov/publications/2010/presentations/TAC2010_Overview.pdf

Dang, H. T., & Owczarzak, K. (2008). Overview of the TAC 2008 update summarization task. *Text Analysis Conference 2008*, 1–16. https://tac.nist.gov/publications/2008/additional.papers/update_summ_overview08.proceedings.pdf

Dang, H. T., & Owczarzak, K. (2009). Overview of TAC 2009 Summarization Track. *Proceedings of the Text Analysis Conference*, 1–25. http://www.nist.gov/tac/publications/2011/presentations/Summarization2011_overview.presentation.pdf

Darwin, C. (1859). The origin of species. In *On the Origin of Species*. https://doi.org/10.4324/9780203509104

Donaway, R. L., Drummey, K. W., & Mather, L. A. (2000). A comparison of rankings produced by summarization evaluation measures. *NAACL-ANLP 2000 Workshop on Automatic Summarization* -, *4*, 69–78. https://doi.org/10.3115/1117575.1117583

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the Association for*

Computing Machinery, *16*(2), 264–285. https://doi.org/10.1145/321510.321519

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, *165*, 113679. https://doi.org/10.1016/j.eswa.2020.113679

Ellouze, S., Jaoua, M., & Belguith, L. H. (2013). An Evaluation Summary Method Based on a Combination of Content and Linguistic Metrics. *Proceedings of the International Conference Recent Advances in Natural Language Processing*, *7–13*(September), 245–251.

Ellouze, S., Jaoua, M., & Belguith, L. H. (2016). Automatic Evaluation of a Summary's Linguistic Quality. In *Lecture Notes in Computer Science* (Vol. 9612, Issue June 2016, pp. 392–400). https://doi.org/10.1007/978-3-319-41754-7_39

Ellouze, S., Jaoua, M., & Belguith, L. H. (2017a). Mix Multiple Features to Evaluate the Content and the Linguistic Quality of Text Summaries. *Journal of Computing and Information Technology*, *25*(2), 149–166. https://doi.org/10.20532/cit.2017.1003398

Ellouze, S., Jaoua, M., & Belguith, L. H. (2017b). Machine Learning Approach to Evaluate MultiLingual Summaries. *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, 47–54. https://doi.org/10.18653/v1/W17-1007

Ermakova, L., Cossu, J. V., & Mothe, J. (2019). A survey on evaluation of summarization methods. *Information Processing & Management*, *56*(5), 1794–1814. https://doi.org/10.1016/j.ipm.2019.04.001

Eshelman, L. J. (1991). The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. *Foundations of Genetic Algorithms*, *1*, 265–283. https://doi.org/10.1016/B978-0-08-050684-5.50020-3

Feng, L., Jansche, M., Huenerfauth, M., & Elhadad, N. (2010). A comparison of features for automatic readability assessment. *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 276–284.

Fiori, A. (Ed.). (2014). *Innovative Document Summarization Techniques*. IGI Global. https://doi.org/10.4018/978-1-4666-5019-0

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, *47*(1), 1–66. https://doi.org/10.1007/s10462-016-9475-9

García-Calderón, M. Á., García-Hernández, R. A., & Ledeneva, Y. (2018). Unsupervised multi-language handwritten text line segmentation. *Journal of Intelligent & Fuzzy Systems*, *34*(5), 2901–2911. https://doi.org/10.3233/JIFS-169476

García-Calderón, M. Á., García-Hernández, R. A., & Ledeneva, Y. (2019). Providing order to the handwritten TLS task: A complexity index. *Journal of Intelligent & Fuzzy Systems*, *36*(5), 4621–4631. https://doi.org/10.3233/JIFS-179013

García-Castro, R. (2013). *Suma de visitas de los pueblos de la Nueva España, 1548-1550* (R.

García-Castro (Ed.); 1st ed., Vol. 64). Universidad Autónoma del Estado de México.

Gelbukh, A., & Calvo, H. (2018). *Automatic Syntactic Analysis Based on Selectional Preferences* (Vol. 765). Springer International Publishing. https://doi.org/10.1007/978-3-319-74054-6

Giannakopoulos, G., & Karkaletsis, V. (2011). AutoSummENG and MeMoG in Evaluating Guided Summaries. *Proceedings of AESOP Text Analysis Conference*, 1–10.

Giannakopoulos, G., Karkaletsis, V., Vouros, G., & Stamatopoulos, P. (2008). Summarization system evaluation revisited: N-Gram Graphs. *ACM Transactions on Speech and Language Processing*, *5*(3), 1–39. https://doi.org/10.1145/1410358.1410359

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Co.

Google. (2021). *Multi-Class Neural Networks: Softmax*. Google Developers. https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax

Gurney, K. (1997). *An Introduction to Neural Networks*. ROUTLEDGE.

Gutierrez-Vasques, X., & Mijangos, V. (2019). Productivity and Predictability for Measuring Morphological Complexity. *Entropy*, *22*(1), 48. https://doi.org/10.3390/e22010048

Härdle, W. K., & Simar, L. (2015). Canonical Correlation Analysis. In *Applied Multivariate Statistical Analysis* (pp. 443–454). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-45171-7_16

Hay, J. C., Murray, A. E., Rosenblatt, F., Stieber, A., & Wolf, R. A. (1960). *Mark I Perceptron Operators' Manual*.

He, T., Chen, J., Ma, L., Gui, Z., Li, F., Shao, W., & Wang, Q. (2008). ROUGE-C: A fully automated evaluation method for multi-document summarization. *2008 IEEE International Conference on Granular Computing*, 269–274. https://doi.org/10.1109/GRC.2008.4664680

Heaton, J. (2008). *Introduction to Neural Networks with Java* (2nd edn.). Heaton Research Inc.

Hershey, J. R., & Olsen, P. A. (2007). Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, *1*(6), IV-317-IV–320. https://doi.org/10.1109/ICASSP.2007.366913

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (Illustrate). MIT Press.

Hotelling, H. (1935). The most predictable criterion. *Journal of Educational Psychology*, *26*(2), 139–142. https://doi.org/10.1037/h0058165

Hovy, E. (2015). Text Summarization. In R. Mitkov (Ed.), *The Oxford Handbook of Computational Linguistics 2nd edition* (2nd edn, pp. 1–24). Oxford University Press.

https://doi.org/10.1093/oxfordhb/9780199573691.013.008

Hovy, E., Lin, C.-Y., & Zhou, L. (2005). Evaluating DUC 2005 using Basic Elements. *Proceedings of the DUC-2005 Workshop*, 1–6.

Hovy, E., Lin, C.-Y., Zhou, L., & Fukumoto, J. (2006). Automated Summarization Evaluation with Basic Elements. *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 899–902.

Jain, A. K., Jianchang, M., & Mohiuddin, K. M. (1996). Artificial neural networks: a tutorial. *Computer*, *29*(3), 31–44. https://doi.org/10.1109/2.485891

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, *186*(1007), 453–461. https://doi.org/10.1098/rspa.1946.0056

Jing, H., Barzilay, R., McKeown, K., & Elhadad, M. (1998). Summarization Evaluation Methods: Experiments and Analysis. *AAAI Symposium on Intelligent Summarization*, 51–59.

Jones, K. S., & Galliers, J. R. (1995). Evaluating Natural Language Processing Systems. In K. S. Jones & J. R. Galliers (Eds.), *Lecture Notes in Artificial Intelligence* (Vol. 1083). Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0027470

Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrika*, *30*(1/2), 81–93. https://doi.org/10.2307/2332226

Kettunen, K. (2014). Can Type-Token Ratio be Used to Show Morphological Complexity of Languages? *Journal of Quantitative Linguistics*, *21*(3), 223–245. https://doi.org/10.1080/09296174.2014.911506

Kolda, T. G. (1998). *Limited-Memory Matrix Methods with Applications* [University of Maryland]. https://drum.lib.umd.edu/handle/1903/483

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Krose, B., & van der Smagt, P. (1996). *An introduction to neural networks* (Eight edn.). University of Amsterdam. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.493

Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., & Steinbrecher, M. (2013). *Computational Intelligence: A Methodological Introduction* (D. Gries, O. Hazzan, & F. B. Schneider (Eds.); Second Edi). Springer London. https://doi.org/10.1007/978-1-4471-5013-8

Kullback, S. (1978). *Information Theory and Statistics* (2nd ed.). Dover Publications, Inc.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Kumar, U., Kumar, V., & Kapur, J. N. (1986). Normalized measures of entropy. *International*

Journal of General Systems, *12*(1), 55–69. https://doi.org/10.1080/03081078608934927

LDE. (2021). *Guide for Determining Text Complexity: Kindergarten through Grade 12 Overview*. https://www.louisianabelieves.com/docs/default-source/teacher-toolbox-resources/guide---how-to-determine-text-complexity-grades-k-12.pdf?sfvrsn=7

Ledeneva, Y., & García-Hernández, R. A. (2013). *Automatic Text Summarization with Maximal Frequent Sequences* (1st edn.). Autonomous University of the State of Mexico.

Ledeneva, Y., & García-Hernández, R. A. (2017). *Automatic Generation of Text Summaries: Challenges, proposals and experiments* (1st edn.). Autonomous University of the State of Mexico.

Ledeneva, Y., García-Hernández, R. A., & Gelbukh, A. (2014). Graph Ranking on Maximal Frequent Sequences for Single Extractive Text Summarization. In *International Conference on Intelligent Text Processing and Computational Linguistics* (Vol. 8404, pp. 466–480). https://doi.org/10.1007/978-3-642-54903-8_39

Lin, C.-Y. (2004). Looking for a Few Good Metrics: ROUGE and its Evaluation. *NTCIR Workshop, June*, 1–8.

Lin, C.-Y., Cao, G., Gao, J., & Nie, J. (2006). An information-theoretic approach to automatic evaluation of summaries. *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 463–470. https://doi.org/10.3115/1220835.1220894

Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using N-gram co-occurrence statistics. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*, *1*(June), 71–78. https://doi.org/10.3115/1073445.1073465

Lin, C.-Y., & Hovy, E. (2002). Manual and automatic evaluation of summaries. *Proceedings of the ACL-02 Workshop on Automatic Summarization* -, *4*(July), 45–51. https://doi.org/10.3115/1118162.1118168

Lin, C. Y. (2001). *SEE: Summary Evaluation Environment* (p. 49). Information Sciences Institute/University of Southern California. http://www1.cs.columbia.edu/nlp/tides/SEEManual.pdf

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. *Proceedings of the Workshop on Text Summarization Branches out (WAS 2004)*, *1*, 25–26.

Lin, J. (1991). Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, *37*(1), 145–151. https://doi.org/10.1109/18.61115

Liu, B. (2015). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press. https://doi.org/10.1017/CBO9781139084789

Liu, Z., & Zhou, J. (2020). Introduction to Graph Neural Networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *14*(2), 127. https://doi.org/10.2200/S00980ED1V01Y202001AIM045

Lloret, E., Plaza, L., & Aker, A. (2018). The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, *52*(1), 101–148. https://doi.org/10.1007/s10579-017-9399-2

Louis, A., & Nenkova, A. (2013). Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, *39*(2), 267–300. https://doi.org/10.1162/COLI_a_00123

Louis, A., & Nenkova, A. (2009). Automatically evaluating content selection in summarization without human models. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1 - EMNLP '09*, *1*(August), 306. https://doi.org/10.3115/1699510.1699550

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, *1*(4), 309–317. https://doi.org/10.1147/rd.14.0309

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, *2*(2), 159–165. https://doi.org/10.1147/rd.22.0159

Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T., & Sundheim, B. (1999). The TIPSTER SUMMAC Text Summarization Evaluation. *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, *1*(1), 77–85.

Mani, I., Klein, G., House, D., Hirschman, L., Firmin, T., & Sundheim, B. (2002). SUMMAC: a text summarization evaluation. *Natural Language Engineering*, *8*(01), 43–68. https://doi.org/10.1017/S1351324901002741

Manning, C. D., Ragahvan, P., & Schutze, H. (2009). *An Introduction to Information Retrieval* (2nd edn., Vol. 21, Issue 8). Cambridge University Press. https://doi.org/10.1109/LPT.2009.2020494

Manning, C. D., & Schutze, H. (1999). Foundations of statistical natural language processing. In *ACM SIGMOD Record* (Second Edi, Vol. 31, Issue 3). MIT Press. https://doi.org/10.1145/601858.601867

Marques de Sá, J. P. (2001). *Pattern Recognition: Concepts, Methods and Applications* (1st Editio). Springer Berlin Heidelberg GmbH.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Melanie, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.

Mendoza, G. A. M., Ledeneva, Y., & Garcia-Hernandez, R. A. (2021). *Detection of main ideas and production of summaries in English, Spanish, Portuguese an Russian. 60 years of research*. Alfaomega Grupo Editor, S.A. de C.V. and Universidad Autónoma del Estado de México. http://ri.uaemex.mx/handle/20.500.11799/112111

Minsky, M., & Papert, S. (1969). Perceptrons, Expanded Edition An Introduction to

Computational Geometry. In *MIT Press*.

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems* (Addison-Wesley (Ed.); 2nd edn). Pearson Education Inc.

Nenkova, A., & Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. *Proceedings of HLT-NAACL*, 145–152.

Nenkova, A., Passonneau, R., & McKeown, K. (2007). The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation. *ACM Transactions on Speech and Language Processing*, *4*(2), 4-es. https://doi.org/10.1145/1233912.1233913

Neri-Mendoza, V., Ledeneva, Y., & García-Hernández, R. A. (2020). Unsupervised extractive multi-document text summarization using a genetic algorithm. *Journal of Intelligent & Fuzzy Systems*, 1–12. https://doi.org/10.3233/JIFS-179900

Over, P. (2001). Introduction to DUC-2001: an Intrinsic Evaluation of Generic News Text Summarization Systems. *Proceedings of the Document Understanding Conference*, 1–53.

Over, P., Dang, H., & Harman, D. (2007). DUC in context. *Information Processing and Management*, *43*(6), 1506–1520. https://doi.org/10.1016/j.ipm.2007.01.019

Over, P., & Ligget, W. (2002). Introduction to DUC-2002: an Intrinsic Evaluation of Generic News Text Summarization Systems. *Proceedings of the Document Understanding Conference*, 1–48. http://duc.nist.gov/pubs/2002slides/overview.02.pdf

Owczarzak, K., & Dang, H. T. (2011). Overview of the TAC 2011 Summarization Track: Guided task and AESOP task. *Proceedings of the Text Analysis Conference (TAC 2011)*, 1–35. https://tac.nist.gov/publications/2011/presentations/Summarization2011_overview.presentation.pdf

Paliwal, M., & Kumar, U. A. (2009). Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, *36*(1), 2–17. https://doi.org/10.1016/j.eswa.2007.10.005

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, July*, 311–218. https://doi.org/10.3115/1073083.1073135

Pardo, T. A. S., Antiqueira, L., Nunes, M. das G. V., Oliveira, O. N., & da Fontoura, L. C. (2006). Modeling and Evaluating Summaries Using Complex Networks. In *Computational Processing of the Portuguese Language: 7th International Workshop, PROPOR 2006, Itatiaia, Brazil, May 13-17, 2006. Proceedings* (pp. 1–10). https://doi.org/10.1007/11751984_1

Pastra, K., & Saggion, H. (2003). Colouring summaries BLEU. *Evalinitiatives '03 Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: Are Evaluation Methods, Metrics and Resources Reusable?*, 35–42.

https://doi.org/10.3115/1641396.1641402

Pearson, K. (1895). VII. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, *58*(347–352), 240–242. https://doi.org/10.1098/rspl.1895.0041

Phillips, I. T., & Chhabra, A. K. (1999). Empirical performance evaluation of graphics recognition systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(9), 849–870. https://doi.org/10.1109/34.790427

Ponce, P. (2010). *Inteligencia artificial con aplicaciones a la ingenieria* (A. Herrera (Ed.); First Edit). Alfaomega.

Ponti, M., Kittler, J., Riva, M., Campos, T. de, & Zor, C. (2017). A decision cognizant Kullback–Leibler divergence. *Pattern Recognition*, *61*, 470–478. https://doi.org/10.1016/j.patcog.2016.08.018

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *40*(3), 211–218. https://doi.org/10.1108/00330330610681286

Porter, M. F. (2001). *Snowball: A language for stemming algorithms*. https://snowballstem.org/texts/introduction.html

Priddy, K. L., & Keller, P. E. (2005). *Artificial Neural Networks: An Introduction*. SPIE, International Society for Optical Engineering. https://doi.org/10.1117/3.633187

Ptak, R., Żygadło, B., & Unold, O. (2017). Projection–Based Text Line Segmentation with a Variable Threshold. *International Journal of Applied Mathematics and Computer Science*, *27*(1), 195–206. https://doi.org/10.1515/amcs-2017-0014

Radev, D. R., Teufel, S., Saggion, H., Lam, W., Blitzer, J., Qi, H., Celebi, A., Liu, D., & Drabek, E. (2003). Evaluation challenges in large-scale document summarization. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*, *1*(1), 375–382. https://doi.org/10.3115/1075096.1075144

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for Activation Functions. *ArXiv*, 1–13. http://arxiv.org/abs/1710.05941

Rankel, P. A., Conroy, J. M., & Schlesinger, J. D. (2012). Better metrics to automatically predict the quality of a text summary. *Algorithms*, *5*(4), 398–420. https://doi.org/10.3390/a5040398

Rathee, S., & Ratnoo, S. (2020). Feature selection using multi-objective CHC genetic algorithm. *Procedia Computer Science*, *167*, 1656–1664. https://doi.org/10.1016/j.procs.2020.03.376

Rino, L. H. M., Pardo, T. A. S., Silla, C. N., Kaestner, C. A. A., & Pombo, M. (2004). A Comparison of Automatic Summarizers of Texts in Brazilian Portuguese. In *Brazilian Symposium on Artificial Intelligence* (Vol. 3171, pp. 235–244). https://doi.org/10.1007/978-3-540-28645-5_24

Robert, C. P., & Casella, G. (2004). *Monte Carlo Statistical Methods* (2nd ed.). Springer New York. https://doi.org/10.1007/978-1-4757-4145-2

Rojas-Simon, J. (2017). *Cálculo de Topline para la generación automática de resúmenes usando algoritmos genéticos* [Autonomous University of the State of Mexico]. http://ri.uaemex.mx/handle/20.500.11799/67369

Rojas-Simon, J., Ledeneva, Y., & Garcia-Hernandez, R. A. (2022). *Evaluation of Text Summaries Based on Linear Optimization of Content Metrics* (Vol. 1048). Springer International Publishing. https://doi.org/10.1007/978-3-031-07214-7

Rojas-Simón, J., Ledeneva, Y., & García-Hernández, R. A. (2021). Evaluation of text summaries without human references based on the linear optimization of content metrics using a genetic algorithm. *Expert Systems with Applications*, *167*, 113827. https://doi.org/10.1016/j.eswa.2020.113827

Rojas-Simón, J., Ledeneva, Y., & García-Hernández, R. A. (2023). ¿Pudo la Inteligencia Artificial ayudar a los médicos con información oportuna durante la pandemia de COVID-19? *Komputer Sapiens*, *CV*(I), 54–65.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. https://doi.org/10.1037/h0042519

Saabni, R., Asi, A., & El-Sana, J. (2014). Text line extraction for historical document images. *Pattern Recognition Letters*, *35*(1), 23–33. https://doi.org/10.1016/j.patrec.2013.07.007

Saggion, H., & Poibeau, T. (2013). Automatic Text Summarization: Past, Present and Future. *Theory and Applications of Natural Language Processing*, *1*(1), 3–21. https://doi.org/10.1007/978-3-642-28569-1_1

Saggion, H., Radev, D., Teufel, S., & Lam, W. (2002). Meta-evaluation of Summaries in a Cross-lingual Environment using Content-based Metrics. *Proceedings of the International Conference on Computational Linguistics (COLING'02)*, 849–855. https://doi.org/10.3115/1072228.1072301

Sai, A. B., Mohankumar, A. K., & Khapra, M. M. (2023). A Survey of Evaluation Metrics Used for NLG Systems. *ACM Computing Surveys*, *55*(2), 1–39. https://doi.org/10.1145/3485766

Salton, G. (1989). Automatic text processing: The Transformation, Analysis, and Retrieval of Information by Computer. In *Addison-Wesley*. Addison-Wesley Publishing Co.

Salton, G., & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, *24*(5), 11. https://doi.org/10.1016/0306-4573(88)90021-0

Salton, G., Wong, A., & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

Sánchez-Juárez, A. S. (2016). Amoxcalli. Un análisis sobre la dimensión ontológica de los códices en los archivos, bibliotecas y museos. *ICOFOM Study Series*, *44*, 57–68. https://doi.org/10.4000/iss.676

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical*

Journal, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Sharma, S., Sharma, S., & Anidhya, A. (2017). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, *4*(12), 310–316.

Sidorov, G. (2013). *Non-linear Construction of N-Grams in Computational Linguistics: Syntactic, Filtered, and Generalized N-Grams* (1st edn.). Sociedad Mexicana de Inteligencia Artificial. http://www.cic.ipn.mx/~sidorov/Nonlinear_construction_of_n_grams.pdf

Sidorov, G. (2019). *Syntactic n-grams in Computational Linguistics* (1st edn.). Springer International Publishing. https://doi.org/10.1007/978-3-030-14771-6

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas*, *18*(3), 491–504. https://doi.org/10.13053/cys-18-3-2043

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14. http://arxiv.org/abs/1409.1556

Sivanandam, S. N., & Deepa, S. N. (2008). Introduction to Genetic Algorithms. In *Springer-Verlag Berlin Heidelberg*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73190-0

Smith, E. A., & Senter, R. J. (1967). Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories (6570th)*, 1–14.

Spearman, C. (1904). The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, *15*(1), 72. https://doi.org/10.2307/1412159

Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., & Alaei, A. (2013). ICDAR 2013 Handwriting Segmentation Contest. *2013 12th International Conference on Document Analysis and Recognition*, 1402–1406. https://doi.org/10.1109/ICDAR.2013.283

Steinberger, J., & Ježek, K. (2009). Evaluation Measures for Text Summarization. *Computing and Informatics*, *28*, 251–275.

Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *91*(8), 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Tang, B., & He, H. (2016). FSMJ: Feature selection with maximum Jensen-Shannon divergence for text categorization. *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, *1*(September 2016), 3143–3148. https://doi.org/10.1109/WCICA.2016.7578786

Torres-Moreno, J.-M. (2014). *Automatic Text Summarization* (J.-C. Pomerol (Ed.); First edit). ISTE Ltd and John Wiley & Sons, Inc.

Torres-Moreno, J.-M., Saggion, H., Da Cunha, I., SanJuan, E., & Velázquez-Morales, P. (2010). Summary Evaluation with and without References. *Polibits*, *42*, 13–19.

https://doi.org/10.17562/PB-42-2

Valy, D., Verleysen, M., & Sok, K. (2017). Line segmentation for grayscale text images of khmer palm leaf manuscripts. *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 1–6. https://doi.org/10.1109/IPTA.2017.8310097

Vania, C., & Lopez, A. (2017). From Characters to Words to in Between: Do We Capture Morphology? *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, *1*, 2016–2027. https://doi.org/10.18653/v1/P17-1184

Vázquez, E., García-Hernández, R. A., & Ledeneva, Y. (2018). Sentence Features Relevance for Extractive Text Summarization using Genetic Algorithms. *Journal of Intelligent & Fuzzy Systems*, *35*(1), 353–365. https://doi.org/10.3233/JIFS-169594

Villegas, M., Puigcerver, J., Toselli, A. H., Sánchez, J. A., & Vidal, E. (2016). Overview of the imageCLEF 2016 handwritten scanned document retrieval task. *CEUR Workshop Proceedings*, *1609*, 233–253.

Williamson, G. L., Fitzgerald, J., & Stenner, A. J. (2013). The Common Core State Standards' Quantitative Text Complexity Trajectory: Figuring Out How Much Complexity Is Enough. *Educational Researcher*, *42*(2), 59–69. https://doi.org/10.3102/0013189X12466695

Xie, L., Zeng, J., Kruger, U., Wang, X., & Geluk, J. (2015). Fault detection in dynamic systems using the Kullback–Leibler divergence. *Control Engineering Practice*, *43*, 39–48. https://doi.org/10.1016/j.conengprac.2015.05.010

Yan, X., Song, D., & Li, X. (2006). Concept-based document readability in domain specific information retrieval. *Proceedings of the 15th ACM International Conference on Information and Knowledge Management - CIKM '06*, 540. https://doi.org/10.1145/1183614.1183692

Yuang, C. T., Banchs, R. E., & Siong, C. E. (2012). An Empirical Evaluation of Stop Word Removal in Statistical Machine Translation. *Computational Linguistics*, 30–37.

Zamanian, M., & Heydari, P. (2012). Readability of Texts: State of the Art. *Theory and Practice in Language Studies*, *2*(1), 43–53. https://doi.org/10.4304/tpls.2.1.43-53

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley Press, Inc. https://doi.org/10.2307/409735