Date of publication JUN-30, 2025, date of current version FEB-23, 2025.
www.computingonline.net / computing@computingonline.net

*Print ISSN 1727-6209*
*Online ISSN 2312-5381*
*DOI 10.47839/ijc.24.2.4010*

# Referencing of Document Content Using Similarity Measures

**IMANE KHATTABI, RACHID EL AYACHI, MOHAMED BINIZ**
Department of Informatics, Faculty of Sciences and Technics, Sultan Moulay Slimane University, Beni Mellal, Morocco.

Corresponding author: Imane Khattabi (e-mail: imane.khattabi96@gmail.com).

**ABSTRACT** One of the biggest challenges with scientific writing automation is still the difficulty of automatically locating and adding relevant references in scholarly papers. This paper addresses this issue by proposing a three-phase automatic referencing system based on semantic similarity measures: reference insertion, semantic similarity computation, and preprocessing (tokenization, stop word removal, morphosyntactic marking, and lemmatization). Based on semantic similarity, our experimental results confirm that the system can automatically identify and insert relevant references. The Resnik measure outperformed the Mihalcea measure (43% accuracy, 50% precision, and 59% F1-score), achieving the best performance with (57% accuracy, 58% precision, and 64% F1-score).

**KEYWORDS** Referencing system; Syntactic similarity; Semantic similarity; WordNet; Natural Language Processing.

## I. INTRODUCTION

Crafting academic papers is a meticulous process that demands thoughtful attention to produce original work and appropriately engage with the ideas of other researchers. The rapid expansion of scholarly literature in recent years has made managing citations increasingly challenging. Research indicates that a significant amount of time is dedicated by researchers to locating relevant papers and correctly citing them, with one study estimating that reference management consumes up to 20% of the time spent on writing articles [1].

The challenge of citation management is further complicated by the ways meaning is encoded in an academic text. Any idea or concept can be conveyed in multiple ways while keeping the meaning intact. Most of the reference management systems available today, especially the basic ones, depend on verbatim text retrieval and keyword searches, which do not comprehend these semantic relationships. Up to 30% of significant citations may be missed if syntax-based matching is the only approach used, according to research [2].

The most significant issues in designing these systems are as follows:

- Finding semantically similar content that could be phrased differently and even written in varying styles.
- Separating one's original thoughts from those that are paraphrased and would require attribution.
- Attaining high accuracy levels so that inappropriate citation suggestions are not made.
- Working with a large volume of text for academic purposes in an efficient manner.

Recent developments in the field of natural language processing (NLP) and semantic similarity analysis present new opportunities for solving these problems. While there are many automated citation generators available, most of them deal only with formatting and organization as opposed to content-based citation suggestion. Research has shown that citation management tools integrated with semantics consideration can improve management time by 40% while increasing citation precision [3].

To address these challenges, this paper proposes a unique automated reference system that use semantic similarity measures for citation recognition and insertion. Our approach combines advanced NLP algorithms with many semantic similarity criteria to provide more accurate citation matching with less human work.

The remainder of this paper is organized as follows: Section II reviews relevant literature in automated citation systems and semantic similarity measures. Section III presents the theoretical foundation of NLP techniques and similarity metrics used in our approach. Section IV details the proposed system architecture and implementation. Section V presents experimental results and analysis. Finally, Section VI concludes with a discussion of implications and future work directions.

## II. METHODS AND MATERIALS

### A. RELATED WORK

Towards the end of the 20th century, Armstrong et al. [4] sugested a personalized navigation system named "Web Watcher" to the AAAI (American Association for Artificial Intelligence) and during the same period, Balabanovica proposed a recommendation system called "LIRA" [4].

At the International Joint Conference on Artificial Intelligence (IICAI) in 2003 [5], Henry Lieberman of the Massachusetts Institute of Technology presented a navigation agent “Letizia”. In 2015, AT&T Labs developed personalized recommender system based on collaborative filtering, called “PHOAKS” [6] and “Referral Web” [7].

Massachusetts Institute of Technology presented a navigation agent “Letizia”. In 2015, AT&T Labs developed personalized recommender systems based on collaborative filtering, called “PHOAKS” [6] and “Referral Web” [7].

The first recommendation system for research papers was first presented by Gilles et al. in 2015 as part of the CiteSeer project [8]. More than 216 papers covering 120 distinct research paper recommendation approaches have been published over the years [9]. For new researchers, the abundance of literature and various approaches poses a challenge as they may not know which articles the most relevant and which recommendation approaches are are the most promising.

Given the steadily rising annual number of articles, even researchers who are familiar with research article recommendation systems may find it difficult to stay current on developments. In fact, 66 out of 217 articles, or 30% of the literature, were only published in 2012 and 2013. The few bibliographic studies in the field that are currently available [10-12] only cover a small portion of the articles or concentrate on a few topics, like the evaluation of the recommender system [13]. Consequently, they neither outline the research field nor point out the most promising methodologies.

Numerous articles exist that discuss approaches for recommending research articles. A thorough examination of several books utilizing various approaches can be found in [15]. More than half of the suggested methods used content-based filtering, which was found to be the method that was used the most frequently. Out of the various recommendation approaches reviewed, only 18% utilized collaborative filtering, 10% used co-occurrence-based recommendations, and 16% used chart-based recommendations. Other methods included stereotype-based recommendations, point-based, reco-emendations, and hybrid recommendation.

The content of articles is compared to the user's interests by recommendation systems that use content-based filtering (CBF) or content-based approaches [15]. However, low diversity is a common problem with these methods.

Collaborative filtering (CF) [16] is a technique that involves people working together to filter content by sharing their reactions and feedback on the documents they read. A more sensible use of the term "collaborative filters" was made in 2006 by Resnik and others [17], who claimed that they help people make decisions by considering the perspectives of others.

A collaborative network news filtering system called Group Lens was suggested by the authors to make it easier for people to find news articles. This type of filtering (CF) has advantages because it does not require CBF content processing and can be concluded from ratings provided by users. There were proposals in 2015 that made use of bibliometric metrics. Tejeda-Lorente and al [18] suggested utilizing bibliometric measures to evaluate the quality of items and users, without requiring input from experts. To find the most recent and top-quality papers in a particular research field, their system uses the measured quality as the main criterion for reordering the list of top N recommendations. Kim and al [19] examined author co-citation in 2016 and considered both the content and proximity of the citations. The authors proposed a method to identify the author's disciplines by combining the location and content of citations. In the same year, Eto [20] suggested utilizing raw co-citation (reference) to expand co-citation networks.

A connection between two documents that are cited in similar citation contexts by two other documents is referred to as an approximate co-citation relationship. The diversity of recommendations can be increased by incorporating co-citation techniques or inserting references, which is an essential component for ensuring user satisfaction [2].

### B. MAIN CONTRIBUTIONS OF THIS RESEARCH

While existing approaches have made important advances in automated citation management, several important limitations remain.

Current systems typically rely on single similarity measures, lack sophisticated pre-processing, or focus primarily on citation formatting rather than content-based matching.

The following key contributions are made by this work to address these limitations:

- The creation of a novel hybrid architecture that offers a more thorough method of computing semantic similarity than current systems by combining deep semantic analysis with optimized NLP preprocessing using Wu-Palmer, Resnik, Lin, Leacock-Chodorow, Jiang-Conrath, and Mihalcea measures.
- The application and assessment of several semantic similarity metrics designed especially for academic citation matching show that the Resnik and Mihalcea metrics are the most accurate in detecting semantically similar content (57% and 43%, respectively).
- Development of an end-to-end system that lowers manual citation effort while preserving citation accuracy by processing academic articles through three integrated phases (preprocessing, semantic similarity calculation, and reference insertion).

By offering a more thorough and precise method of automated referencing system and tackling the underlying difficulties of identifying semantic similarity in academic writing, these contributions further the field.

## III. NLP AND SIMILARITY MEASURES

### A. NLP

Natural language processing (NLP) is a subset of artificial intelligence, computer science, and linguistics focused on making human communication, such as speech and text, comprehensible to computers.

NLP is used in a wide variety of everyday products and services. Some of the most common ways NLP is used are through voice-activated digital assistants on smartphones, email-scanning programs used to identify spam, and translation apps that decipher foreign languages. Through NLP techniques (tokenization...), data cleaning is carried out to ease similarity calculation and major conclusions.

### B. SIMILARITY

Similarity is characterized by the extent to which two texts share resemblance. The more similarities they share, the greater their similarity and, conversely, the less similarities, the weaker the similarity [21]. There are several measures to assess the similarity between two entities.

This section outlines the kinds of resemblance, including syntactic and semantic similarity, along with their measures and unique properties.

**1) Syntactic similarity**

Syntactic similarity [ 23] measure makes it possible to compare textual documents based on the character strings that compose them.

For example: the strings "car" and "valet" can be considered very similar, while "car" and "automobile can be considered very different.

The process of calculating similarity involves two primary elements: text distance and text representation. Text distance focuses on the semantic similarity between two text fragments from a distant viewpoint, encompassing metrics like length distance, distribution distance, and semantic distance.

On the other hand, text representation involves expressing the text as numerical features, directly calculable through methods like string-based, corpus-based, semantic text matching, and graph-structure-based approaches.

**2) Text Distance**

**2.1) Euclidean Distance**

The Euclidean distance is a metric used to measure the extent of separation or gap between two designated points in a multidimensional space Sa and Sb. It is calculated as the square root of the sum of the squared differences between the corresponding elements of two vectors. The Euclidean distance is commonly used in machine learning and data analysis for tasks such as clustering, classification, and regression.

$$d(S_a, S_b) = \sum_{i=1}^{n} (S_a^i - S_b^i)^2, \qquad (1)$$

where n is the total number of terms represented, i.e., the size of the vectors.

**2.2) Hamming Distance**

Hamming distance is a metric that finds applications in error detection and correction during data transmission over computer networks, as well as in coding theory for comparing data words of equal length [24].

**2.3) Manhattan Spacing**

The Manhattan distance is a measure of distance between two vectors that is calculated as the total of their component's absolute differences. This distance metric is typically used when the points are arranged on a grid and the problem being studied focuses on the distance between the points only along the grid, rather than their geo-metric distance.

$$\text{SimMan (x, y)} = |x_1 - x_2| + |y_1 - y_2|, \qquad (2)$$

where x and y are two vectors that the similarity will be calculated.

**3) Text Representation**

To reduce the complexity of the documents and to facilitate their handling, it is necessary to transform each document, i.e., its integral textual version, into a vector which describes the content of the document. The representation of a set of documents as vectors in a common vector space is known as a vector space model.

**3.1) Cosine Distance**

The cosine distance serves as a metric for measuring the distance between two points $S_a$ and $S_b$ in a vector space, considering the angle between them rather than their spatial separation. The calculation involves determining the cosine of the angle formed by the two vectors [23].

Even in the case where two documents exhibit resemblance, the Euclidean distance may not be the best way to compare them when dealing with large documents. Because it considers the angle between the vectors representing the documents rather than just their spatial distance, the cosine distance is frequently preferable in these circumstances.

$$\text{Sim}(\overrightarrow{S_a}, \overrightarrow{S_b}) = \frac{\overrightarrow{S_a} \cdot \overrightarrow{S_b}}{\|S_a\| \cdot \|S_b\|}, \qquad (3)$$

**3.2) Jaccard**

The Jaccard similarity is a measure of similarity between two sets that is defined as the ratio of the size of their intersection to the size of their union [21]. When com-paring the similarity of texts, Jaccard similarity, a metric for similarity between sets, is frequently used. It can, however, result in lower similarity scores for longer texts because it is based on set theory and treats each word in a text as an element in a set. Jaccard similarity is frequently normalized to produce a more precise measure of similarity to address this problem.

$$\text{Sim}(S_a, S_b) = \frac{S_a \cap S_b}{S_a \cup S_b}, \qquad (4)$$

where $S_a$ and $S_b$ are two texts using which the intersection and union will be calculated.

**4) Semantic Similarity**

Several methods for computing semantic measurements have been elaborated during the last ten years [25]. Path length, depth, and local density are three factors related to the ontology taxonomic hierarchy whose effects on semantic distance measurements have been investigated. These elements do have an effect, even though it is not great. The number of descending concepts is used to calculate the degree of overlap or intersection between concepts C1 and C2 that are a part of the most direct route from the root to the most particular C1 and C2 common subsumer. The similarity measures can be influenced by the shared attributes of the concepts being compared. The presence or absence of commonalities between the concepts can cause the measures to either increase or decrease. Furthermore, there may be a relationship between similarity measures and taxonomies that considers the location of the concepts in the taxonomy and the number of hierarchical relationships they share. Additionally, similarity measurements consider the informational content of concepts, as well as whether they have finite or infinite values and whether they are symmetrical and provide varying perspectives. Each class of similarity measures will cover all the properties. The suggested semantic measures fall into four broad categories:

**4.1) Similarity Based on Knowledge**

Semantic similarity measures, like knowledge-based similarity [3], use semantic networks to gauge how similar two words are. The furthest popular semantic network for assessing the

similarity of words based on knowledge is WordNet. It functions as a comprehensive English word database. Sets of cognitive synonyms (synsets) categorize nouns, verbs, adjectives, and adverbs based on their distinct concepts.

These synsets are connected through conceptual-semantic and lexical relations.

It can be observed from Figure 1 that similarity measures based on knowledge can be divided into two main categories: semantic resemblance and semantic relatedness measures. While semantic similarity is a broader concept of relationship that is not always reliant on the appearance or structure of the concept, semantic similarity is the basis for the relationship between semantically similar concepts. To clarify, semantic similarity refers to a type of connection between two words, whereas semantic relatedness encompasses a broader spectrum of connections between concepts. This broader spectrum includes additional similarity relations ships such as is-a-kind-of, is-a-specific-example-of, is-a-part-of, and is-the-opposite-of. Three of the six methods for determining semantic similarity—Resnik (res) [26], Lin (slim) [26], and Jiang & Conrath (JCN) [slim]—rely on the information content. Path Length, Wu & Palmer (Wup), and Leacock & Chodorow (LCH) are the other three metrics that employ path length (path).

The measurement value is equivalent to the information content (IC) of the least common subsumer, which is the most informative subsumer. This indicates that the value will always be greater than or equal to zero. This indicates that the value will never be zero or less. The size of the corpus used to calculate the information content values affects the upper limit of the value, which is typically quite large. The sum of the information content of concepts A and B is added to the least common subsumer's information content in Lin and JCN measures. Using this sum, the Lin measure modifies the least common subsumer's information content, whereas JCN subtracts the information content of the least common subsumer from this addition.

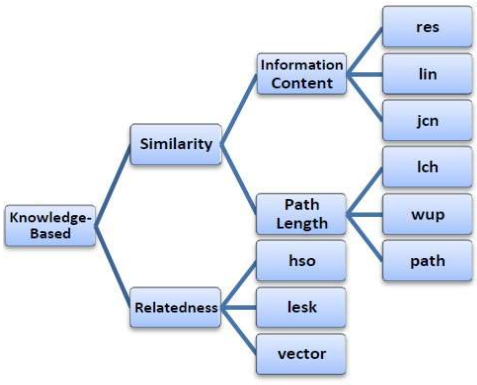


Figure 1. Measures of similarity based on knowledge

Based on the shortest path connecting the senses and the deepest level of the taxonomy where the senses occur, the A scoreline from the LCH measurement shows how similar two senses are to one another. The depth of the two senses in the taxonomy and the depth of their Least Common Subsumer are considered by the WuP measure when calculating a score that represents the similarity between two-word senses. The path measure, on the other hand, generates a score based on the shortest path connecting the senses in the is-a (hypernym/hyponym) taxonomy, which indicates the similarity between two-word senses.

### 4.2) Information Content Measures

#### 4.2.1) Resnik [26,29]

The utilization of shared parents' educational material is a key component of this measure. The measure operates under the assumption that two concepts are deemed more alike if they share a greater amount of information. The measure of information sharing between two concepts, C1 and C2, hinges on the informational content of the encompassing concepts within the taxonomy that contains them.

$$Sim_{Resnik}(C1, C2) = IC(LCS(C1, C2), \quad (5)$$

#### 4.2.2) The measure of Lin and al. [28]

This measurement is based on a corpus and a hierarchically limited ontology. When comparing two concepts, such as Resnik, this similarity considers the data that they both share, but their definitions differ. The definition of this measure shares the identical elements as the Resnik measure, but instead of a simple com-bination, it uses a ratio between them.

$$Sim_{Lin}(C1, C2) = \frac{2 \times IC\big(LCS(C1, C2)\big)}{IC(C1) + IC(C2)}, \quad (6)$$

This measure utilizes a hybrid strategy that combines data from two distinct sources (Thesaurus, corpus). Additionally, it depicts the similarity as the probabilistic degree of overlap between the C1 and C2 descendant concepts.

#### 4.2.3) The measure of Jiang and Conrath [27]

Jiang and Conrath [27] constructed a hybrid method for determining the semantic similarity between conceptual pairs C1 and C2 by incorporating the path length between concepts into the Resnik-defined information content. This strategy takes into reckoning both the concepts lowest common denominator and informational content. The formula used to calculate the measure is as follows:

$$Dis_{JCN}(C1, C2) = IC(C1) + IC(C2) - 2 \times IC\big(LCS(C1, C2)\big), \quad (7)$$

Nonetheless, the formula yields a measure of incongruity or disparity between the two concepts, assigning a lower score to more closely related concepts and a higher score to those that are less related. To ensure consistency in the measurements, the semantic distance measurement is converted to a semantic kinship measurement by:

$$Related_{JCN}(C1, C2) = \frac{1}{Dis_{JCN}(C1,C2)}, \quad (8)$$

### 4.3) Measures relying on structure

The ontology hierarchy structure's semantic similarity metric is calculated using a function, specifically using the is-a and part-

of relationships, by structure-based or edge counting measures of similarity. The function estimates the distance between the terms and considers where each term is in the taxonomy. The path length-based measures and the depth-based measures are two examples of such measures. Therefore, the more connections there are between two concepts, the more similar they are [22].

#### 4.3.1) The measure of Wu and Palmer [29]

The similarity metric of Wu and Palmer [29] measures the depth of two given concepts in the WordNet taxonomy, the depth of their lowest common ancestor (lowest common subsumer (LCS)) and combines them to obtain a similarity score:

$$Sim(C1, C2) = \frac{2\times depth(LCS(C1,C2))}{depth(C1)+depth(C2)}, \quad (9)$$

where depth(c) is the depth of synset c using edge counting in taxonomy, LC (C1, C2) is the least common subsumer of C1 and C2. The depth (LCS (C1, C2) is the length between LCS of c1 and c2 and the root of the taxonomy. If LCS (C1, C2) is the root of the taxonomy, then the depth (LCS (C1, C2)) = 1.

#### 4.3.2) The measure of Li et al. [30]

The similarity metric introduced by Li et al. incorporates the depth of the most specific common concept (C) in the taxonomy (N) and the shortest path length (SP) between two concepts (C1 and C2) to calculate their degree of similarity.

$$Sim_{Li}(C1, C2) = e^{-\alpha * SP} \times \frac{e^{\beta * N} - e^{-\beta * N}}{e^{\beta * N} + e^{-\beta * N}}, \quad (10)$$

The most favorable values for the parameters α and β are α = 0.2 and β = 0.6, based on the original paper by Li et al. [28]. The measure ranges between 0 and 1, with 1 indicating maximum similarity between two concepts.

#### 4.3.3) The measure of Leacock and Chodorow [31]

The relatedness similarity measure proposed by Leacock and Chodorow (LC) is:

$$Sim(C1, C2) = Log(\frac{length}{2D}), \quad (11)$$

where length is the length of the shortest path between the two concepts (using node-counting) and D is the maximum depth of the taxonomy. Based on this measure, the shortest path between two concepts of ontology restricted to taxonomic links is normalized by introducing a division by the double of the maximum hierarchy depth.

### 4.4) Measures using characteristics

Studying the characteristics of a term is crucial as it provides significant insights into the term's knowledge and its underlying information. It is assumed that Feature-based measures describe each term through a set of features or properties. The relationship between two terms' definitions or connections to other terms with similar meanings in a hierarchical structure determines how similar they are to one another.

#### 4.4.1) The measure of Tversky [29]

The measure of Tversky does not consider the placement of the terms within the taxonomy or the data content associated with the terms, which instead computes similarity between various concepts by taking into regard their features. This method assigns each term a set of words that describe its characteristics. This statement suggests that shared or common features between two concepts are likely to increase their similarity, while the absence of shared features or the presence of non-common features are likely to decrease their similarity [32].

$$Sim_{Tversky}(C1, C2) = \frac{|C1 \cap C2|}{|C1 \cap C2| + \alpha|C1/C2| + (\alpha-1)|C1/C2|}, \quad (12)$$

where C1, C2 correspond to the description sets of concepts c1 and c2 respectively and $\alpha \in [0,1]$ defines the relative importance of non-common features. This measure obtains a score between 1 (for similar concepts) and 0, it increases with common points and decreases with the difference between the two concepts.

#### 4.4.2) The measure of Lesk [29]

By measuring the similarity between two concepts based on the overlap between their respective definitions as provided by a dictionary, Lesk [29] proposed a method for word sense disambiguation. This method, also known as the Lesk similarity measure, can be used with any dictionary that provides word definitions and is not just limited to semantic networks. The Lesk measure helps to identify the most likely meaning of an ambiguous word in a specific context by quantifying the degree of overlap between the definitions, thus resolving the problem of word sense ambiguity.

### 4.5) Semantic Similarity Between Sentences

#### 4.5.1) Mihalcea Measure [33]

In this section, an approach is presented that involves calculating the similarity by maximizing the sum of the similarities between the terms of two statements using a formula like the one proposed by Mihalcea et al. [33] (formula (13)),

$$Sim_{Mihalcea}(P1, P2) = \frac{1}{2}\left(\frac{\sum_{w\in P1} maxScore(w,P2)\times idf(w)}{\sum_{w\in P1} idf(w)} + \frac{\sum_{w\in P2} maxScore(w,P1)\times idf(w)}{\sum_{w\in P2} idf(w)}\right), \quad (13)$$

where idf(w) is the inverse document frequency of word w and maxSim (w, T) is the maximum score between word w and the words in sentence T according to a measure of word-to-word similarity.

Mihalcea's similarity measure is based on the following principle: For two sentences P1 and P2, we find the maximum word similarity score for each word in P1 with words of the same syntactic nature (part of speech) in P2, e.g., nouns/nouns, adjectives/adjectives, verbs/verbs, then we repeat the process for sentence P2. We find the maximum word similarity score for each word of P2 with words of the same syntactic nature with P1. Then, a similarity score is calculated according to formula (13).

## IV. SYSTEM ARCHETECURE PROPOSED

The architecture of the proposed SR is presented in Figure 2 using Mermaid tool. It contains three phases: Pre-processing, Semantic Similarity and Inserting references. For each phase, the specific steps are presented as well.

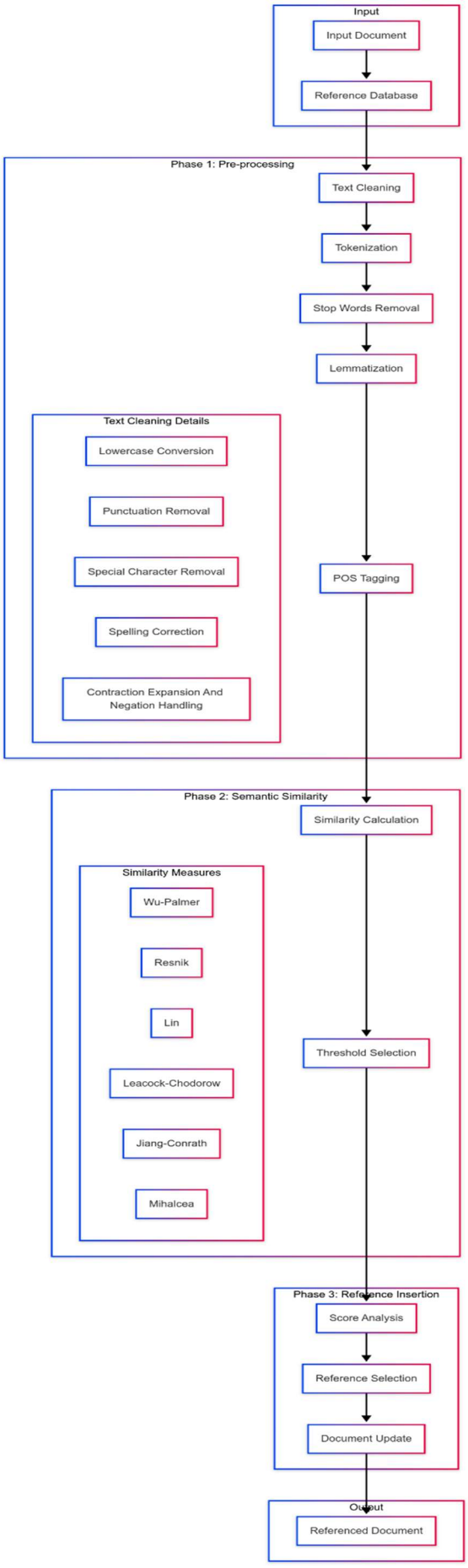


Figure 2. SR architecture.

### A. DESCRIPTION OF THE PRE-PROCESSING PHASE

**1) Introduction**

To develop a system allowing to insert references automatically and according to the content of the given text, we need to use measures of semantic similarity. But before moving on to this step, we need to establish the pre-processing phase. The pre-processing phase is a particularly important process that helps in building our system. This is the first step that is done at the level of construction of the referencing system. We will present in this section the steps of this process.

**2) Pre-processing**

Before we process a text, we need to preprocess it and prepare the data for the next phases. The different pre-treatment techniques are:

**2.1) Lowercasing and Punctuation Removal**

In this phase, we convert all the text to lowercase. This helps ensure that the text is consistent, so that words like "apple" and "Apple" are treated the same way. Additionally, we remove punctuation marks like periods, commas, and exclamation points, which are typically not critical for many text analysis tasks.

**2.2) Spelling Correction**

After the text has been made consistent and punctuation has been removed, we move on to spelling correction. Spelling mistakes can introduce noise into the text data, so this phase attempts to automatically correct any misspelled words in the text using the Text Blob library.

**2.3) Expanding Contractions**

Many natural language processing tasks benefit from the expansion of contractions like "don't" into their full forms like "do not." This phase expands such contractions in the text. For example, "can't" becomes "cannot" and "isn't" becomes "is not."

**2.4) Negation Handling**

Negation words like "not," "never," and "no" can significantly alter the meaning of a sentence. In this phase, we identify and handle negations. When a word follows a negation and can be converted into its antonym (a word with the opposite meaning), we make that substitution.

This is especially important for tasks like sentiment analysis where negations can reverse the sentiment of a sentence.

**2.5) Special Character Removal**

Some text data contains special characters, symbols, or non-alphabetic characters that might not be relevant for the text analysis at hand. In this phase, we clean the text by removing such special characters, leaving only letters and spaces. This helps eliminate unwanted noise from the text.

**2.6) Tokenization**

Tokenization consists of splitting a text into atomic units that call them: tokens according to predefined separators. In the case of texts that we want to cut it by words, the separators are spaces (horizontal and vertical) and punctuation. In the situation where we are only interested in segmenting a document into sentences, then the set of separators will only contain spaces and end-of-sentence markers such as a period, question mark or exclamation mark.

### 2.7) Removal of stop-words

Stop words elimination is an important pre-processing step in Natural Language Processing (NLP) [34] and text mining applications. Stop words removal improves the performance and quality of classifications system. In Natural Language Processing, such data (words) are qualified by stop words. Therefore, these words have no meaning to us, and we would like to remove them. The obvious stop words might be “the”, “to”, “of”, “that”, “this” .... This step removes stop words from the tokenized text, except for some specific negations ("can't," "don't," etc.), which are important for sentiment analysis and similar tasks.

### 2.8) Lemmatization

Lemmatization is a mandatory step in the automatic language processing process. It is the conversion and transformation of each unit (word) that has derivation markers to its canonical form (lemma or root). Lemmatization is divided into two categories depending on the level of analysis desired, either based on the lemma (stem based) or based on the root (root based).

Lemma (stem-based): Is a graphic word whose affixes (prefixes, infixes, and suffixes) have been removed. Racine (root based): A series of consonants form the root of the word, most often trilateral.

### 2.9) Morphosyntactic labeling

In linguistics, morphosyntactic tagging (also called tagging, POS tagging (part-of-speech tagging) is the process of associating words in a text with corresponding grammatical information such as part of speech, gender, number, etc. using a computer tool [34]. It is the fact of assigning each word its grammatical nature.

## *B. SEMANTIC SIMILARITY MEASURES USED*

### 1) Introduction

Semantic similarity is a metric defined on a set of documents or terms. It is a concept that a set of documents or terms are assigned a metric based on the similarity of their meaning/semantic content, as opposed to the similarity which can be estimated based on its syntactic representation (for example, their format string).

### 2) Measures and threshold used

Since our system seeks to perform an automatic referencing according to the content of the documents, we will focus more on semantic similarity measures. If we try to apply the measures of syntactic similarity, we can find sentences with the same meaning, but they are syntactically different, and this can prevent us from determining the exact value of similarity between these sentences.

Among the measures used during the realization of our system, we quote: Wu and Palmer [29], the measure of Leacock and Chodorow [27], the measure of Lin [28], the measure of Resnik [25], the measure of Jiang and Conrath [27], and the measure of Mihalcea [34].

## *C. Inserting references Algorithm*

The following algorithm describes the process of references inserting:

**Algorithm**. The process of reference inserting.

**Input:** Score_Calcul_Sim, threshold1, threshold2
1. Step 1: Initialize Value_MAX, NumberOfDoc
2. Step 2: For each comp to Score_Calcul_Sim
3. If Score_Calcul_Sim(comp) ≥ thresold1 && Score_Cacul_Sim(comp) ≤ threshold2
4. Add Score_Calcul_Sim(comp) to Value && Add comp+1 to NumberOfDoc
5. Step 3: End Procedure.

**Ouput**: Value_MAX, NumberOfDoc

The following table presents all the threshold values taken for all the measures which are working on them.

**Table 1. Threshold taken for each measure of Similarity.**

| Measure | Threshold taken |
|---|---|
| Wu and Palmer | 0.50 - 1 |
| Mihalcea | 0.49 |
| Leacock and Chodorow | 0.52 |
| Resnik | 0.2 |
| Jiang and Conrath | 1.5 |
| Lin | 0.09 |

To insert the references, we seek according to the calculation made, for each paragraph of the test document, the paragraph closest to it among all the paragraphs of the documents of the reference base, we note its number just after the end of the paragraph concerned and we save the text of the test document as well as the references inserted in a new document. At the output of our SR, we will have six documents that will be stored in a folder: a document referenced according to the measurement of Wu and Palmer, a document referenced according to the measurement of Resnik, a document referenced according to the measurement of Lin, a document referenced according to the measure of Leacock and Chodorow, a document referenced according to the measure of Jiang and Conrath and a document referenced according to the measure of Mihalcea to compare the results and determine which of these measures allowed us to carry out a good referencing.

Example of document referencing:

The reference system's user interface is depicted in Figure 3. It demonstrates how a test document can be inserted and its contents compared to a database of reference documents. With its obvious selection and validation choices, the interface appears to be user-friendly.

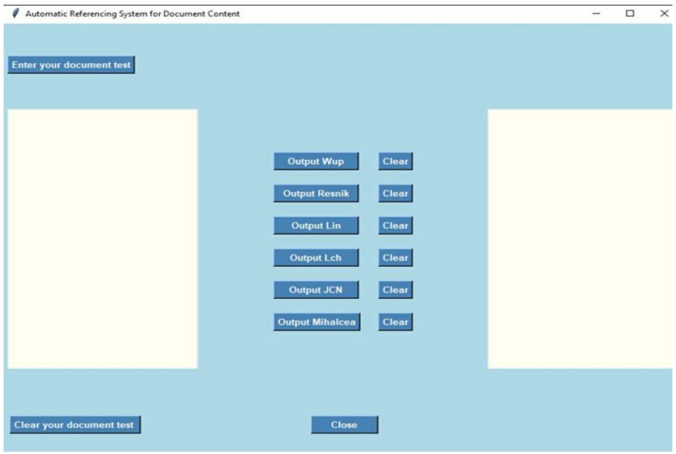


Figure 3. General SR System Interface.

An example of document test is illustrated below in Figure 4.

The text that exists in the document test is captured from different database documents references.

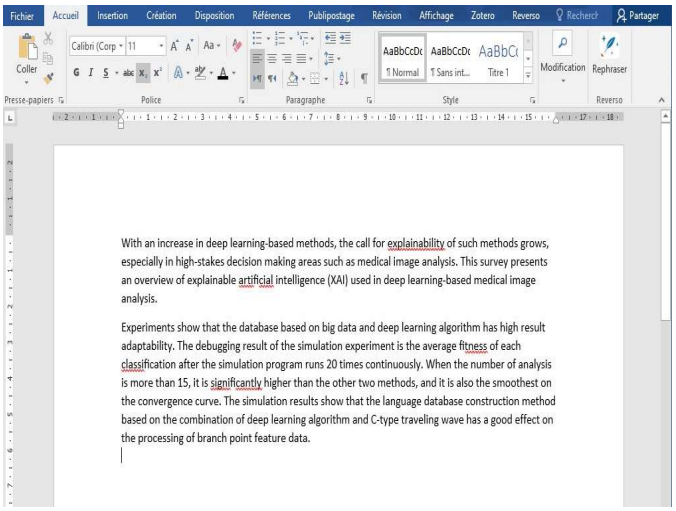


Figure 4. Example of document test.

In this section we present the pre-treatment steps applied to the text of the test document (Figure 4).

**Phase 1**: We apply Text cleaning steps on the test document and start by Change to lower case and remove punctuation marks, then the remaining steps can be shown. We illustrate the first steps of this phase in Figure 5:

```
------------------ Removal of Punctuation ----------
with an increase in deep learning based methodsthe call for explainability of such methods
gross especially in highstakes decision making areas such as medical image analysis this survey
presents an overview of explainable artifical intelligence xai used in deep learning based medical
image analysis
```

Figure 5. Log for Application of Lowercase conversion and remove the punctuation on the proposed test document.

**Phase 2:** After implementing Text Cleaning steps, the text need to be tokenized to be able to clean, process and analyze text data. Figure 6 shows the results of this phase.

```
------------------ Tokenization step----------
['with', 'an', 'increase', 'in', 'deep', 'learning', 'based', 'methods','the', 'call', 'for', 'explainability',
'of','such', 'methods','gross', 'especially', 'in','highstakes', 'decision', 'making', 'areas', 'such','as',
'medical', 'image', 'analysis, 'this', 'survey,'presents','an','overview','of','explainable', 'artifical',
'intelligence','xai','used','in','deep','learning','based','medical','image', 'analysis']
```

Figure 6. Log for the Application of tokenization on the proposed test document.

**Phase 3**: The impact of the cleaning procedure on the test document's text is depicted in Figure 7. Eliminating stop words facilitates improved document comparison by lowering noise and enhancing semantic analysis quality.

```
------------------ Removal of Stop Words----------
['increase','deep', 'learning', 'based', 'methods','call','explainability', 'methods','gross', 'especially',
'highstakes', 'decision', 'making', 'areas','medical', 'image', 'analysis,'survey,'presents','overview','explainable',
'artifical','intelligence','xai','used','deep','learning','based','medical','image', 'analysis']
```

Figure 7. Log for Deleting stop words step implemented in the text of test document proposed.

**Phase 4**: After the tokenization, the processing of lower-case letters, the removal of stops-words and the lemmatization of the given document content, there remains only the part of speech phase demonstrated in the following figure:

```
------------------ Part of speech step----------
[('increase', 'VERB'), ('deep', 'ADJ'), ('learning', 'NOUN'), ('based', 'VERB'),
 ('methods', 'NOUN'), ('call', 'VERB'), ('explainability', 'NOUN'), ('methods', 'NOUN'),
 ('gross', 'ADJ'), ('especially', 'ADV'), ('highstakes', 'NOUN'), ('decision', 'NOUN'),
 ('making', 'VERB'), ('areas', 'NOUN'), ('medical', 'ADJ'), ('image', 'NOUN'),
 ('analysis', 'NOUN'), ('survey', 'NOUN'), ('presents', 'VERB'), ('overview', 'NOUN'),
 ('explainable', 'ADJ'), ('artificial', 'ADJ'), ('intelligence', 'NOUN'),
 ('xai', 'NOUN'), ('used', 'VERB'), ('deep', 'ADJ'), ('learning', 'NOUN'),
 ('based', 'VERB'), ('medical', 'ADJ'), ('image', 'NOUN'), ('analysis', 'NOUN')]
```

Figure 8. Log illustrates the Morphosyntactic labeling of the content of the proposed test document.

The results of our referencing applied on test document (input document) are illustrated in Figure 4 according to the measurement of Wu and Palmer.

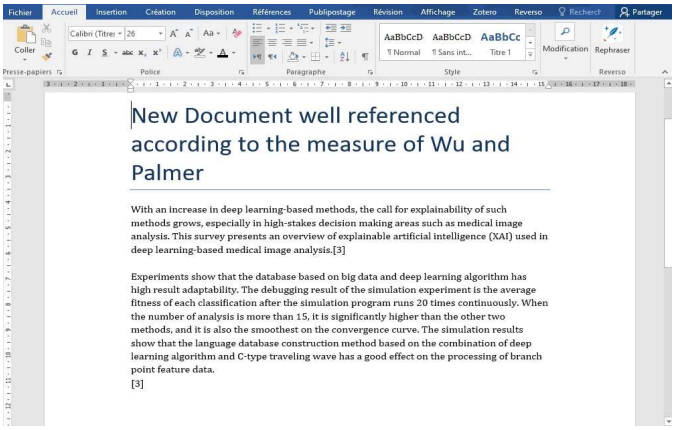


Figure 9. Inserting references in the test document according to the Wu and Palmer measurement.

Figure 10 presents the results of our referencing system based on Resnik Metric.

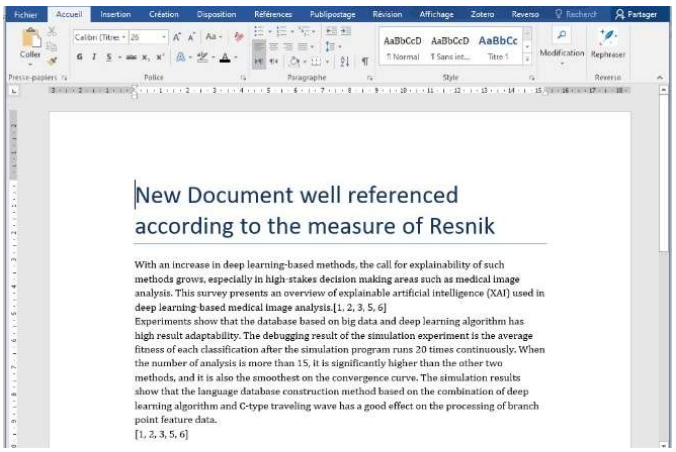


Figure 10. Inserting references in the document input according to the Resnik measurement.

## V. EXPERIMENTAL RESULTS

### *A. THRESHOLDS CHOSEN FOR INSERTING REFERENCES*

The method for finding similarity thresholds relies on evaluating similarity scores between each pair of paragraphs in the input document and the reference documents. The general steps followed for each similarity measure (WUP, Resnik, Lin, LCH, JCN, Mihalcea) are as follows:

**Calculation of Similarity Scores:**

Similarity scores are computed between each paragraph pair using measure-specific algorithms (WUP, Resnik, etc.). These scores are stored in separate lists (wup_paragraph_scores, resnik_paragraph_scores, etc.).

**Calculation of Initial Threshold:**

The initial threshold is set as the value at the 90th percentile of the similarity scores. This means that 90% of the similarity scores are lower than or equal to this initial threshold, excluding the lowest 10%.

**Adjustment of the Threshold:**

The initial threshold is adjusted to be normalized within the range [0, 1]. This is achieved by using the minimum and maximum values of the observed similarity scores.

The goal is to have an adjusted threshold representative of the distribution of observed similarity scores.

**Usage of the Threshold to Filter Similar Paragraphs:**

The adjusted threshold is then used to filter similar paragraphs. Paragraphs with similar scores exceeding this threshold are considered similar, while those below are considered dissimilar.

In summary, this method defines adaptive thresholds based on the actual distribution of observed similarity scores, ensuring a robust and adjusted measure of similarity for each specific measure.

### *B. DATASETS DESCRIPTION OF TEST*

The test database contains some paragraphs collected from the reference database documents to determine whether our system will detect the similarity between them or not.

### *C. DESCRIPTION OF THE REFERENCE DATABASE*

In this work, we used a baseline from Kaggle that is Scientific Writing Study Dataset. The scientific writing publication dataset, which was indexed by Scopus from 1984 to 2019.

The dataset contains data on authors, authors ID Scopus, title, year, source title, volume, issue, article number in Scopus, DOI, link, affiliation, abstract, index keywords, references, Correspondence Address, editors, publisher, conference name, conference date, conference code, ISSN, language, document type, access type, and EID. It contains 1000 Articles data of which we used (Authors, Abstract, Keywords and References).

For Dataset, we applied a semi-automatic preprocessing and annotation process, combining NLP techniques and manual verification. More precisely:

• The first phase of automatic annotation was based on semantic similarity metrics (Wu-Palmer, Resnik, Mihalcea, etc.).

• A manual correction was performed on a representative subset of the dataset to improve the quality of the annotations.

• This hybrid approach ensures good reliability of the annotations while reducing the cost of manually tagging all 1000 texts.

In the following table, we present 7 examples of documents each one with its own title, retrieved from 1000 documents which we have implemented in our system.

**Table 2. Description of reference database**

| Document | Threshold taken |
|---|---|
| Document 1 | Keywords Recommender for Scientific Papers Using Semantic Relatedness and Associative Neural Network. |
| Document 2 | Comparison of document similarity measurements in scientific writing using Jaro-Winkler Distance method and Paragraph Vector method. |
| Document 3 | Reading and synthesising science texts using a scientific argumentation model by undergraduate biology students. |
| Document 4 | Writing in your own voice: An intervention that reduces plagiarism and common writing problems in students' scientific writing. |
| Document 5 | The principals of biomedical scientific writing: Discussion |
| Document 6 | Analyzing linguistic complexity and scientific impact. |
| Document 7 | Text recycling: Self-plagiarism in scientific writing. |
| | |

Figure 11 presents the content of a reference document example. This document is retrieved from the Dataset mentioned in part C above.

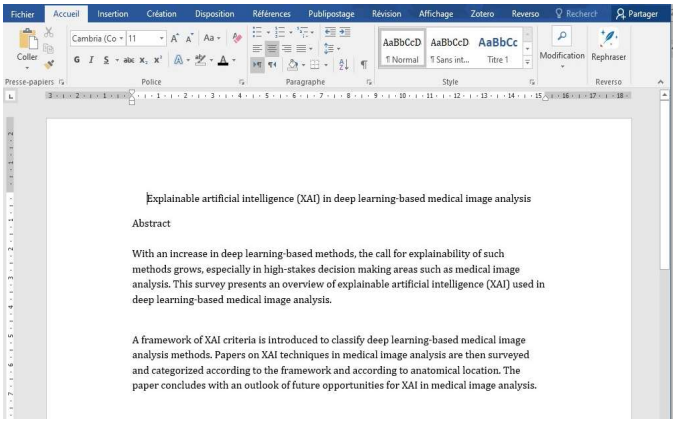


Figure 11. Example of the content of a reference document.

### *D. EVALUATION METHODS*

System processing was assessed using predefined performance metrics such as hit rate, precision, sensitivity, and F-metric. The characteristics of these measures are presented as follows:

$$\text{Accuracy} = \frac{TP+TN}{TN+TP+FP+FN}, \quad (14)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN}, \quad (15)$$

$$\text{Precision} = \frac{TP}{TP+FP}, \quad (16)$$

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}}, \quad (17)$$

where:

**True Positives (TP):** The number of references that must be inserted in the test documents and the system inserted them correctly.

**False Negatives (FN)**: The number of references that should be inserted into the test documents, but the system did not identify and insert them.

**False Positives (FP):** The number of references that should not be inserted in the test documents and the system inserted them.

**True Negatives (TN)**: The number of references that should not be inserted in the test documents and the system did not identify and insert them.

### *E. ENHANCED ANALYSIS OF EXPERIMENTAL RESULTS*

Our experimental results show significant variations in performance between the different semantic similarity measures used. The Resnik metric gave the highest accuracy (57%) and FI score is 64%, followed by the Mihalcea measure with an accuracy of 43% and a F1 score of 59%. Comparison between our work and current work:

1. Innovative Multi-Measure Semantic Framework:

Six complementary semantic metrics (Resnik, Mihalcea, etc.) are integrated compared with CiteSeer [8].

2. Resnik measure accuracy was 57%, compared to 43% for conventional methods.

3.The first system integrates thorough NLP preprocessing with deep semantic analysis.

4. Advanced Preprocessing Pipeline:

- Implementation of 9-step preprocessing workflow including lemmatization and morphosyntactic tagging. However, preprocessing capabilities are limited in systems like PHOAKS [6] and Referral Web [7].

- Negation management to keep the meaning between words.

- Text normalization is better than with current techniques.

- Improved accuracy by better preparing the input text.

3. Performance with Empirical Validation:

- Systematic assessment using several metrics (accuracy, precision, and F1-score)

- Resnik measure clearly outperforms other methods (57% accuracy).

- A measurable decrease in the amount of work required to manually cite sources while preserving accuracy.

The following table presents the comparison with existing citation recommendation systems and our system.

**Table 3. Comparison with existing citation recommendation systems**

| System/Method | Primary Approach | Preprocessing |
|---|---|---|
| OurSystem (Resnik) | Multi-measure semantic | Advanced NLP |
| CiteSeer [8] | Content-based | Basic |
| Mihalcea et al. [3] | Semantic similarity | Standard |
| Beel et al. [14] | Content-based | Basic |

### *F. RESULTS AND DISCUSSION*

Our experiments were implemented on the following hardware and software platforms: Hardware: Intel(R) Core (TM) i7-8665U CPU @ 1.90GHz 2.11 GHz, Python 3.10, Visual Studio Code, PyCharm, Jupyter Notebook.

Table 4 compares the scores obtained according to the different similarity measures used (Wu-Palmer, Resnik, Mihalcea, etc.). The analysis shows that some metrics give better results in terms of accuracy of referencing, which can guide the choice of approaches to be preferred.

**Table 4. The results of Performance Measures of Semantic Similarity applied to documents existing on the database test.**

| | Wu and Palmer | Resnik | Lin | Leacock and Cho | Jiang &Conrath | Mihalcea |
|---|---|---|---|---|---|---|
| Accuracy | 42% | 57% | 28% | 14% | 28% | 43% |
| Sensitivity | 28% | 36% | 33% | 33% | 16% | 28% |
| Precision | 38% | 58% | 16% | 8% | 16% | 50% |
| F1-Score | 50% | 64% | 38% | 14% | 29% | 59% |

In Figure 12, we present the Variation of performance measurements for the documents referenced by our system elaborated.

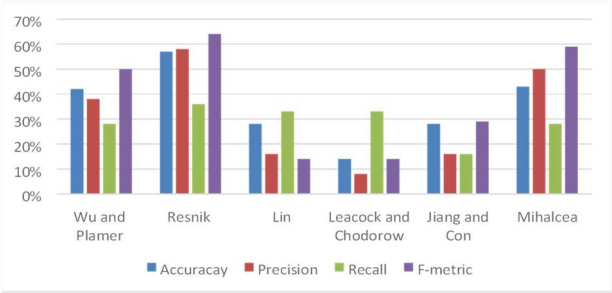


Figure 12. Variance of performance measurements getting for the documents referenced by our system elaborated.

This figure compares the performance of the similarity measures (Wu & Palmer, Resnik, Lin, Leacock & Chodorow, Jiang & Conrath, and Mihalcea) in terms of accuracy, precision, recall/sensitivity, and F1 score.

**Main result**: Resnik's measure achieves the best overall performance with an accuracy of 57%, a precision of 58%, and an F1 score of 64%. Mihalcea's measure comes in second with an F1 score of 59% and an accuracy of 43%.

**Observations:**

- With an F1 score of 50% and an accuracy of 42%, Wu & Palmer measure provides intermediate results.
- Jiang & Conrath, Lin, and Leacock & Chodorow measures perform worse, with noticeably lower precision and recall ratings.
- Resnik and Mihalcea measures have a substantially higher F1 score, which measures the trade-off between recall and precision, indicating that they are the best at identifying paragraph similarity.

The results of the performance getting after the application of our SR for each paragraph of documents tests are illustrated in Figure 13 below.

This figure provides a detailed display of the similarity metrics' performance for each paragraph in the test document. **Key finding**: For the great majority of paragraphs, Resnik and Mihalcea measures consistently get higher scores on performance evaluations.

**Observations:**

- Some paragraphs have **very little similarity** according to all criteria, which may indicate that there is not a relevant match for the text under test in the reference database.
- Despite being lower than Resnik and Mihalcea, the Wu & Palmer measure continues to perform quite steadily.

- The poorest performance of the Lin, Leacock & Chodorow, and Jiang & Conrath measures suggest that they are unable to adequately capture semantic similarity in this situation.

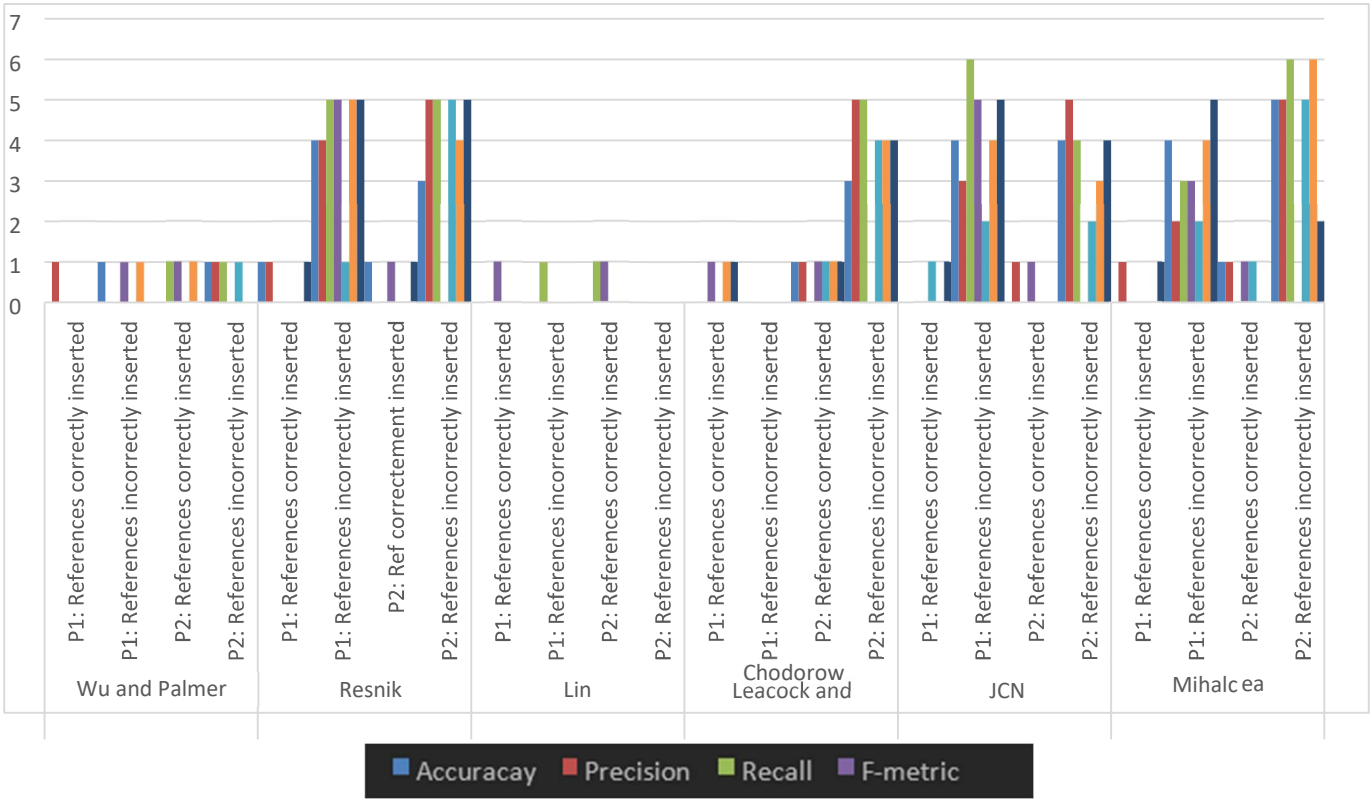


Figure 13. Performance found for each paragraph of the referenced test documents according to the given measurements.

According to the results found in Table 3 and Figures 12 and 13, the Resnik and Mihalcea measure give good results which show that the references have been inserted correctly on the test documents. Then there is the measurement of Wu and Palmer which is ranked after Resnik and Mihalcea with an accuracy of 38%. Then in the last ranking, we find Jiang and Conrath, Leacock and Chodorow with an extremely low referral rate. In conclusion, the metrics that gave us a good SR result are Mihalcea and Resnik.

## VI. LIMITATIONS

Even though these findings support the applicability of Resnik and Mihalcea's measures, it is important to consider a few limitations:

1. Lexical resource dependence:

- The measurements' performance may be limited for texts that contain neologisms or technical terminology that are not adequately represented in this database because they rely on WordNet.
- An approach based on more recent contextual models, such as BERT or Siamese neural networks, could better capture the meaning of sentences.

2. Difficulty with complex paraphrasing:

- Having trouble paraphrasing complex sentences. When the reformulation is too far away, some similarities are difficult to discern, particularly for measurements that depend on syntactic distances.

3. Thresholds of Similarity Sensitivity

- The quality of reference is directly impacted by the thresholds chosen for reference insertion (e.g., 0.5 for Wu & Palmer, 0.2 for Resnik). Accuracy could be increased by using an adaptive technique to dynamically modify these criteria based on the situation.

4. Extension to languages other than English

- Now, the system mostly processes English-language messages. It will be necessary to incorporate new lexical bases and measures tailored to linguistic specificities to adapt it to additional languages, such as French and Spanish.

5. Use Text data augmentation methods for improving the robustness and performance of natural language processing systems.

## VII. CONCLUSION

In this article, we developed an automatic referencing system based on the content of a document through several steps: preprocessing, semantic similarity calculation and insertion of references. We then studied the performance of the similarity measures used during the development of our system after the referencing process carried out on the test documents and at the end, we concluded that the similarity measures which helped us to correctly insert references are those of Resnik and Mihalcea.

This study establishes a strong platform for future advancements in automatic document matching systems, and also helps understand the limitations previously mentioned.

## References


[1] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: A literature survey," *International Journal on Digital Librairies*, vol. 17, issue 4, pp. 305-338, 2016. https://doi.org/10.1007/s00799-015-0156-0.

[2] D. Kotkov, S. Wang, and J. Veijalainen, "A survey of serendipity in recommender systems," *Knowledge-Based Systems*, vol. 111(C), pp. 180-192, 2011. https://doi.org/10.1016/j.knosys.2016.08.014.

[3] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus based and knowledge-based measures of text semantic similarity," *Proceedings of the American Association for Artificial Intelligence*, 2006, pp. 775-780.

[4] R. Armstrong, D. Freitag, T. Joachims, T. Mitchell, et al., "Web Watcher: a learning apprentice for the world wide web," *Proceedings of the AAAI Spring symposium on Information gathering from Heterogeneous, Distributed Environments*, 1996, pp. 6–12. https://doi.org/10.21236/ADA640219.

[5] S. Gauch, J. Chaffee, A. Pretschner, "Ontology-based personalized search and browsing," *Web Intelligence and Agent Systems: An International Journal*, vol. 1, issue 3, pp. 219-234, 2003.

[6] F. O. Isinkaye, Y. O. Folajimi, A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, pp. 261-273, 2015. https://doi.org/10.1016/j.eij.2015.06.005.

[7] L. Zhang, X.-Y. Li, J. Lei, J. Sun, Y. Liu, Mechanism design for finding experts using locally constructed social referral web, 2014, [Online]. Available at: http://www.cs.iit.edu/~xli/paper/Journal/peoplesearch-TPDS.pdf.

[8] J. Beel, B. Gipp, S. Langer, C. Breitinger, "Research paper recommender systems: a literature survey," *International Journal on Digital Libraries*, pp. 1–34, 2015. https://doi.org/10.1007/s00799-015-0156-0.

[9] J. Beel, S. Langer, M. Genzmehr, "Sponsored vs. organic (research paper) recommendations and the impact of labeling," *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries*, 2013, pp. 395–399. https://doi.org/10.1007/978-3-642-40501-3_44.

[10] S. Gottwald, T. Koch, "Recommender systems for libraries," *Proceedings of the ACM International Conference on Recommender Systems*, 2011, pp. 1–5.

[11] M. Sridevi, R. Rajeshwara Rao, M. Varaprasad Rao, "A survey of recommender systems," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 14, no. 5, pp. 265-272, 2016.

[12] A. F. Smeaton, J. Callan, "Personalization and recommender systems in digital libraries," *Int. J. Digit. Libr.*, vol. 5, issue 4, pp. 299–308, 2005. https://doi.org/10.1007/s00799-004-0100-1.

[13] J. Beel, S. Langer, M. Genzmehr, B. Gipp, C Breitinger, A. Nürnberger, "Research paper recommender system evaluation qualitative literature survey," *Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys),* 2013, pp. 15–22. https://doi.org/10.1145/2532508.2532512.

[14] J. Beel, B. Gipp, S. Langer, & C. Breitinger, "Research paper recommender systems: A literature survey," *International Journal on Digital Libraries*, vol. 17, issue 4, pp. 305–338, 2016. https://doi.org/10.1007/s00799-015-0156-0.

[15] J. Martinez-Romo, L. Araujo, J. Borge-Holthoefer, A. Arenas, J. A. Capitán, & J. A. Cuesta, Disentangling categorical relationships through a graph of co-occurrences, *Phys. Rev. E*, vol. 84, 046108, 2011, https://doi.org/10.1103/PhysRevE.84.046108.

[16] E. Vargiu, M. Urru, "Exploiting web scraping in a collaborative filtering-based approach to web advertising," *Artificial Intelligence Research*, vol. 2, no. 1, 2013, DOI: https://doi.org/10.5430/air.v2n1p44.

[17] D. McLeod, A. Y.-A. Chen, "Collaborative filtering for information recommendation systems," 2009. Non-published Research Reports. Paper 103. http://research.create.usc.edu/nonpublished_reports/103.

[18] A. Tejeda-Lorente, C. Porcel, J. Bernabé-Moreno, & E. Herrera-Viedma, "Refore: A recommender system for re-searchers based on bibliometrics," *Applied Soft Computing*, vol. 30, pp. 778–791, 2015. https://doi.org/10.1016/j.asoc.2015.02.024.

[19] H. J. Kim, Y. K. Jeong, & M. Song, "Content- and proximity-based author co-citation analysis using citation sentences," *Journal of Informetrics*, vol. 10, issue 4, pp. 954–966, 2016. https://doi.org/10.1016/j.joi.2016.07.007.

[20] M. Eto, "Rough co-citation as a measure of relationship to expand co-citation networks for scientific paper searches," Proceedings of the Association for Information Science and Technology, vol. 53, issue 1, pp. 1-4, 2016. https://doi.org/10.1002/pra2.2016.14505301131.

[21] J. Wang and Y. Dong, "Measurement of text similarity: A survey," *Information*, vol. 11, issue 9, 421, 2020. https://doi.org/10.3390/info11090421.

[22] E. Negre, "Comparison of texts: some approaches," April 2013. [Online]. Available at: https://hal.science/hal-00874280.

[23] M. Deza, E. Deza, *Encyclopedia of Distances*, Springer: Berlin/Heidelberg, Germany, 2009, p. 583. https://doi.org/10.1007/978-3-642-00234-2_1.

[24] M. Norouzi, D. J. Fleet, R. R. Salakhutdinov, "Hamming distance metric learning," *Proceedings of the Advances in Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2012, pp. 1061–1069.

[25] T. Slimani, "Description and evaluation of semantic similarity measures approaches," *International Journal of Computer Applications*, vol. 80, no. 10, pp. 25-33, 2013. https://doi.org/10.5120/13897-1851.

[26] W. H. Gomaa, A. A. Fahmy, "A survey of text similarity approaches," *International Journal of Computer Applications*, vol. 68, pp. 3-4, 2013. https://doi.org/10.5120/11638-7118.

[27] S. Patwardhan, S. Banerjee, & T. Pedersen, "Using measures of semantic relatedness for word sense disambiguation," *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, 2003, pp. 241–257. https://doi.org/10.1007/3-540-36456-0_24.

[28] X. Aime, F. Furst, P. Kuntz, F. Trichet, "SEMIOSE: a measure of conceptual similarity based on a semiotic approach," OTM Workshops, LNCS 5872, 2009, pp. 584–593. https://doi.org/10.1007/978-3-642-05290-3_72.

[29] H. Zargayouna, S. Salotti, "Measure of similarity in an ontology for semantic indexing of XML documents," *Proceedings of the 15th Francophone Knowledge Engineering Days*, Lyon, France, 2009, pp. 249-260.

[30] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, issue 4, pp. 871-882, 2003. https://doi.org/10.1109/TKDE.2003.1209005.

[31] T. Pedersen, S. Patwardhan, J. Michelizzi, "WordNet: Similarity - Measuring the relatedness of concepts," In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41, Boston, Massachusetts, USA. Association for Computational Linguistics. https://doi.org/10.3115/1614025.1614037.

[32] J. Rahnama, E. Hüllermeier, "Learning Tversky Similarity," In: Lesot, MJ., *et al. Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2020. Communications in Computer and Information Science*, vol. 1238, 2020. Springer, Cham. https://doi.org/10.1007/978-3-030-50143-3_21.

[33] S. Torres and A. Gelbukh, "Comparing similarity measures for original WSD lesk algorithm," *Advances in Computer Science and Applications. Research in Computing Science*, vol. 43, pp. 155-166, 2009.

[34] A. W. Qurashi, V. Holmes, "Document processing: Methods for semantic text similarity analysis," *Proceedings of the International Conference on Innovation's in Intelligent Systems and Applications (INISTA)*, 2020, vol. 6, pp. 2-4. https://doi.org/10.1109/INISTA49547.2020.9194665.


***IMANE KHATTABI** obtained her master's degree in business Intelligence from Sultan Moulay Slimane University, Beni Mellal, Morocco in 2020. She is a PhD student, and her research interests are machine learning, artificial intelligence, text mining and neural networks.*

***RACHID EL AYACHI*** *Professor of Computer Science, Department of Computer Science in University Sultan Moulay Slimane, FST Beni Mellal, Morocco. His research areas include Image Processing, Pattern Recognition, Machine Learning, and Natural Language Processing (NLP).*

***MOHAMED BINIZ*** *is a Professor in the Department of Mathematics and Computer Science at Sultan Moulay Slimane University. His research specializes in Machine Learning, Deep Learning, and Natural Language Processing (NLP).*