10.31341/jios.47.2.10
Open Access
UDC 004.738.5:004.8
Original Scientific Paper

Optimization of the results of a multilingual search engine using a fuzzy recommendation approach

Amine El Hadi *elhadi.amine@gmail.com*
Faculty of Sciences and Technics,
Sultan Moulay Slimane University, Beni Mellal, Morocco

Youness Madani *younesmadani9@gmail.com*
Faculty of Sciences and Technics,
Sultan Moulay Slimane University, Beni Mellal, Morocco

Rachid El Ayachi *rachid.elayachi@usms.ma*
Faculty of Sciences and Technics,
Sultan Moulay Slimane University, Beni Mellal, Morocco

Mohamed Erritali *m.erritali@usms.ma*
Faculty of Sciences and Technics,
Sultan Moulay Slimane University, Beni Mellal, Morocco

Abstract

Search engines are now the main source for information retrieval due to the huge expansion of data on the internet over the last ten years. Providing users with the most relevant results for their queries poses a significant challenge for search engines. Semantic search engines, which go beyond traditional keyword-based searches, have appeared as advanced information retrieval systems to address this problem. These search engines produce more precise and pertinent search results because they understand the meanings of words and their relationships. They play a pivotal role in managing the vast amount of internet data, with a primary aim of enhancing search precision and user satisfaction. However, improving search precision remains as an important goal for natural language processing researchers. The main objective of our research is to improve the search engine results. We present a novel approach for measuring the similarity between a user's query and a list of documents within a search engine. This approach provides a new fuzzy recommendation system using a syntactic and semantic similarity. Our results indicate that our method outperforms several existing approaches from the literature, achieving a high level of accuracy.
Keywords: Semantic similarity, Fuzzy Logic, Search engine, Query reformulation

1. Introduction

With the growth of the Web, a wide range of services are now available on the internet, but information retrieval [1] remains one of the most popular activities. People may

exchange a wide variety of information online since websites are now simpler to make and utilize. However, finding the precise information you need may not be simple due to the huge amount of information available on the Internet.

In other words, because there is a lot of information available, searching for things on the Internet can be difficult. The more information there is, and the more people are looking for it at the same time, the more complicated it becomes.

Therefore, we may say that without search engines, finding information on the internet would be extremely difficult. Search engines are like practical tools that do the following four tasks: First, they collect information-containing web pages. They organize these web sites together in a second step. They also enable links to be made between web pages. Fourth, they encourage queries from the public and use special methods to find the best websites that provide answers.

Today, search engines [31] have become one of the most helpful tools for obtaining useful information from the Internet. Search engines became the most helpful tool for obtaining useful information from the Internet [30]. However, the search results returned by traditional search engines aren't satisfactory. as an example, when looking for news stories about phd students, with traditional searching technologies, we regularly could only get news entries within which the term "PhD students" appears. Those entries which mention the names of scholars but don't use the term "PhD students" directly are going to be passed over. To overcome this problem the Semantic search engines appears where the meaning of web content is made explicit.

One important goal of the semantic web is to make the meaning of data explicit through semantic mark-up, thus enabling simpler access to knowledge contained in heterogeneous information environments, like the web. Semantic search plays a crucial role in realizing this goal, because it promises to provide precise answers to user's queries by taking advantage of the availability of explicit semantics of information.

The rest of this paper is organized as follows: Section 2 presents literature review, in Section 3 we will describe our Research methodology (the spelling checks and correction method, the language detection and query translation and the calculation of the semantic similarity between documents and finally how we obtain the relevant documents using Fuzzy recommendation system). Section 4 presents the experimental results and finally, in Section 5 the conclusion and the perspectives.

2. Literature Review

In recent years, a lot of approaches have been developed for improving the results of search engines, in regard to returning the most relevant documents for the users. In the literature, we found many studies have been presented on getting the most relevant documents by calculating the similarity between documents.

The authors in [2] proposes a new cluster-based information retrieval approach named ICIR (Intelligent Cluster-based Information Retrieval, the approach combines k-means clustering with frequent closed itemset mining to find the most frequent terms in each cluster. The discussed model obtains clusters of documents and obtains

the most frequent terms in each cluster. The most relevant document clusters are then selected with respect to the patterns discovered in each cluster.

Nguyen et al. [3] propose a tri-partite neural language model that leverages explicit knowledge to jointly constrain word, concept, and document representations. The authors employ the model in two retrieval strategies: document re-ranking and query expansion, and they show the effectiveness of their approach in various IR tasks.

Researchers in [4] presented an approach on ontology-based detection mechanism, they succeed by their approach to reduce memory consumption, decreasing the number of resultant documents, and minimizing the search time. in Their approach They calculate a hash value of each document, then by matching them, they calculate an index, which indicates the similarity or not.

Authors in [5] proposed a novel neuro fuzzy approach for semantic textual similarity that uses neural networks and fuzzy logics. They used the remarkable capabilities of the current neural models that extract and convert features associated with text expressed in natural language with the possibilities that fuzzy logic provides for aggregating numerical information and decoding in a personalized way information of numerical nature.

In this article [6] researchers presented a Fuzzy set similarity measure between Fuzzy Words. They use four different kinds of fuzzy set similarity measures, three that are standard ones for type-1 fuzzy sets and another one based on the distance between defuzzified and normalized COGs for type-2 fuzzy sets and examines both the Pearson and Spearman correlations among these different fuzzy set similarity measures on fuzzy words from four of the six categories established in past sentence similarity research.

Suma V. [7] improves retrieval efficiency and accuracy, addressing issues with conventional information retrieval methods. A novel hybrid deep fuzzy hashing technique was presented by the researcher. The hashing method efficiently retrieves information by mapping similar data into correlated binary codes. Deep neural networks and fuzzy logic are used to train the underlying information, which allows the system to efficiently extract the needed data from distributed cloud sources.

In the work of S.H Farhi et al. [8], the graph-based information retrieval system is well-known and frequently used in a variety of applications. The proposed bibliographic information system has been built to process text-based queries and retrieve information through its interface, building upon this graph-based framework. In addition, Joby et al. [9] discussed the problems with using a natural language model to get information from big data sets. Considering the limitations in probabilistic, space vector, and other conventional retrieval models, the proposed study places a particular focus on a natural language-based retrieval system that retrieves pertinent data from massive datasets.

Ontology-Based Semantic Information Retrieval (NOSIR), a novel method that combines feature selection and classification techniques, was introduced by Selvalakshmi et al. [10]. This technique's main goal is to retrieve data from huge datasets. The authors used fuzzy rough set-based feature selection and Latent Dirichlet Allocation-based semantic information retrieval techniques to further improve feature

selection and classification outcomes. The main advantages of the proposed algorithms are the increase in relevancy, ability to handle big data and fast retrieval.

Esposito et al. [11] developed a hybrid query expansion (HQE) approach based on lexical resources and word embeddings for the question-answering system. The question-answering system was predicted from the information retrieval process, and the questions were received from the MultiWordNet before being provided contexts for the document collection. The Word2Vec model was utilized to generate the results, and performance metrics such as accuracy and Mean Reciprocal Rank (MRR) were employed to achieve optimal results.

The retrieval system can extract information from the vast database with the aid of machine learning-based models because they usually perform efficiently and provide better classification results. The authors of [12] evaluate the most recent studies that have been done in a variety of fields, such as the classification of texts, the analysis of medical diseases the classification of user smartphones and images, etc. In comparison to other methods described in the literature, the decision trees (DT) approach improves classification results, as demonstrated in this study, which presents a detailed approach to the algorithm. Authors in [13] introduced a text classification model for BBC news using machine learning algorithms. They discussed the logistic regression, random forest, and K-nearest neighbor algorithms, providing a thorough analysis of each component of the model as well as the metrics for model evaluation. The results demonstrate that the logistic regression classifier, when combined with the TF-IDF Vectorizer feature, achieves the highest accuracy of 97% for the dataset. This algorithm has proven to be the most reliable classifier, particularly for smaller datasets. Ranjitha and Prasad in [14] presented different machine learning techniques like Hadoop map-reduce and naive Bayes classifiers to classify the data. Their demonstration revealed that Gaussian naive Bayes enhances text classification rates when compared to other machine learning approaches.

Authors in [26] utilizes fuzzy ontology to enhance information retrieval systems through query expansion. It involves creating a concept dictionary from a specific domain and external ontology, assigning fuzzy membership using ConceptNet's Global Ontology, and defining fuzzy membership for various semantic relationships. The proposed method calculates membership values based on semantic and ConceptNet edge weights, enabling the identification of related concepts within the domain and expanding queries. Evaluation against various parameters showed improved results compared to previous research in literature.

Table 1 show a comparison between multiple approaches cited in the literature review:

Reference	Year	Methodology	Key Findings
[2]	2018	Cluster-based IR	Classification of most relevant document clusters.
[3]	2018	Neural language model	Ranking and query expansion.
[4]	2020	Ontology based detection mechanism	Optimizing search time and improving results

[5]	2021	Neural network and Fuzzy logic	Improved semantic similarity approach.
[6]	2019	Fuzzy set similarity	Improved similarity measures.
[7]	2020	Hybrid deep fuzzy hashing technique	Improved retrieval efficiency and accuracy.
[8]	2018	Graph-based information retrieval	Process text-based queries and retrieve information through its interface.
[9]	2020	Information retrieval from big data sets	Improved retrieval approach from massive datasets.
[10]	2019	Ontology-based semantic IR using Fuzzy	Proposed a new method to increase relevancy, ability to handle big data and fast retrieval.
[11]	2020	Hybrid query expansion	Proposed a new approach based on lexical resources and word embeddings to improve accuracy and Mean Reciprocal Rank (MRR).
[12]	2021	Decision Tree	Improved text classification rate.
[13]	2020	Logistic Regression	Improved classification rate on BBC news dataset.
[14]	2020	Hadoop map-reduce and Naive Bayes classifiers	Improved results on classification rate using Gaussian Naive Bayes.
[26]	2021	Fuzzy ontology	Improved information retrieval systems through query expansion.

Table 1. Comparison table for the literature papers.

3. Fuzzy Logic

In this section, we will present the main concepts of the fuzzy logic: fuzzy sets, fuzzification, fuzzy Rules inference, and Defuzzification.

The concept of fuzzy logic [18], [19] is quite similar to how humans think and reason. Instead of the binary "true or false" (1 or 0) logic that underlies modern computers, it represents a computing approach that is based on "degrees of truth." Fuzzy logic is generally used to solve issues that don't have an obvious resolution and instead have several shades of gray [28]. Deterministic logic aligns with crisp sets [29] in the field of set theory, whereas fuzzy logic was first proposed in 1965 by Professor L. A. Zadeh at the University of California, Berkeley [15].

Our daily lives are flawlessly intertwined with fuzzy logic and fuzzy ideas, often without our even noticing it. Think about how, in some surveys, we might respond with vague or ambiguous words like "Not Very Satisfied" or "Quite Satisfied," which are effectively fuzzy or ambiguous replies. These replies reveal our level of satisfaction or dissatisfaction with a service or good. Only humans, and not machines, are capable of generating and understanding responses with such nuance. Given that computers can only comprehend '0' or '1,' 'HIGH' or 'LOW,' it is impossible for them to directly respond to survey questions in such a sophisticated way. Crisp or binary data, which form the basis of machine processing, are these forms of data.

3.1. Fuzzy Sets

The concept of a fuzzy set is an extension of the concept of a classic or crisp set. A classical set only takes into account a limited amount of degrees of membership, generally '0' or '1,' where the value of the membership function for an object is either 0 (showing that the object does not belong to the set) or 1 (saying that the object totally belongs to the set).

A fuzzy set, on the other hand, is a generalization of the classical set having a range of 0 to 1. A fuzzy set allows for partial membership from objects, with the degree of membership increasing with the strength of the object's link with the set.

A useful tool for representing objects or members that allows for ambiguity or vagueness is fuzzy sets. For example, in our case we want to get the most relevant documents to the query in an ambiguous way.

3.2. Fuzzification

To apply fuzzy logic to a real-world problem, like document classification in our case, three consecutive steps are required: Fuzzification, fuzzy inference, and defuzzification.

Implementing a fuzzy logic system begins with fuzzification [17]. It involves a transformation from precise to fuzzily defined quantities. Most variables are clear or classic in the real world. These crisp variables must be transformed into fuzzy variables in order to be used in a fuzzy logic system, both as input and output. Fuzzification consists of two essential processes: building membership functions for input and output variables and presenting them using linguistic variables.

Membership functions are crucial for transforming sets of crisp values into fuzzy ones. In practice, there are various types of membership functions, including triangular, trapezoidal, Gaussian, bell-shaped, sigmoidal, and S-curve waveforms.

In order to start the process, a crisp variable needs to be fuzzified. The next step involves applying a specific membership function, such as triangular or trapezoidal function, to this variable. As a result, the variable has a degree of membership that falls between [0,1].

3.3. Fuzzy Rules

The second step of fuzzy logic systems is fuzzy rules [20]. By describing links between input and output variables using linguistic variables and membership functions, they define the logic and decision-making processes in these systems. When there is a lack of precision or uncertainty in the information, fuzzy rules are utilized to make decisions or control actions. A fuzzy rule is a simple IF-THEN rules with a condition and a conclusion. For example, in our work, if semantic similarity is high and syntactic similarity is low then document is relevant. The inputs and the outputs of the rules have fuzzy values.

Fuzzy rules enable fuzzy logic systems to handle complex, real-world problems by incorporating human-like reasoning and decision-making processes that consider

uncertainty and vagueness in data. These rules are commonly used in various applications, such as control systems, expert systems, and decision support systems.

3.4. Defuzzification

Defuzzification [16] is the third stage of a fuzzy logic system (FLS). Most of actions or decisions taken by humans or machines are clear-cut and binary, despite the abundance of fuzzy data we receive on a daily basis. We take binary decisions, and the hardware we use also operates in a binary manner. As a result, as in tasks like classification, we have to transform the fuzzy outputs into crisp values that provide us the solution to our problem.

Defuzzification is the procedure that transforms a fuzzy set into a single valued crisp quantity or a crisp set. The logical union of two or more fuzzy membership functions that are defined inside the context of discourse of the output variable may be the result of a fuzzy process.

There are several defuzzification methods documented in the literature, including the Max-membership principle, the Centroid Method (also known as the center of area or center of gravity), the Weighted Average Method, and the Mean-max membership method (also known as middle-of-maxima).

4. Research Methodology

In this section, we will present the different steps of our work and the methodology of our proposed approach. The aim of our work is to get the most relevant tweets of a query in a search engine using a fuzzy logic system based on semantic and syntactic similarity. To make our proposed approach, we have to implement different steps either in the collection of tweets, the preparation of the tweets for the classification (text preprocessing methods), then classify the tweets according to their class: relevant or not to the query of the user and return the most relevant tweet to the query using the proposed fuzzy approach.

Our system is composed of different steps, we will start by the preparation of the query used by the user (spell check and correction) to get a clean query without any error in the spelling, then and to not treat just the English tweet, we need to translate all the collected tweets to English if this is not the case. To apply our fuzzy logic system we need to calculate the semantic similarity based on n-grams and reinforcement learning and syntactic similarity, the next step is to use these 2 measures as an inputs to our Fuzzy logic system and apply all the different steps of the fuzzy system (Fuzzification, Fuzzy rules and Defuzzification) to find the most relevant tweets, the last stage of our proposed approach is to sort the relevant tweet by the degree of belonging to the relevant class and return these tweets to the user.

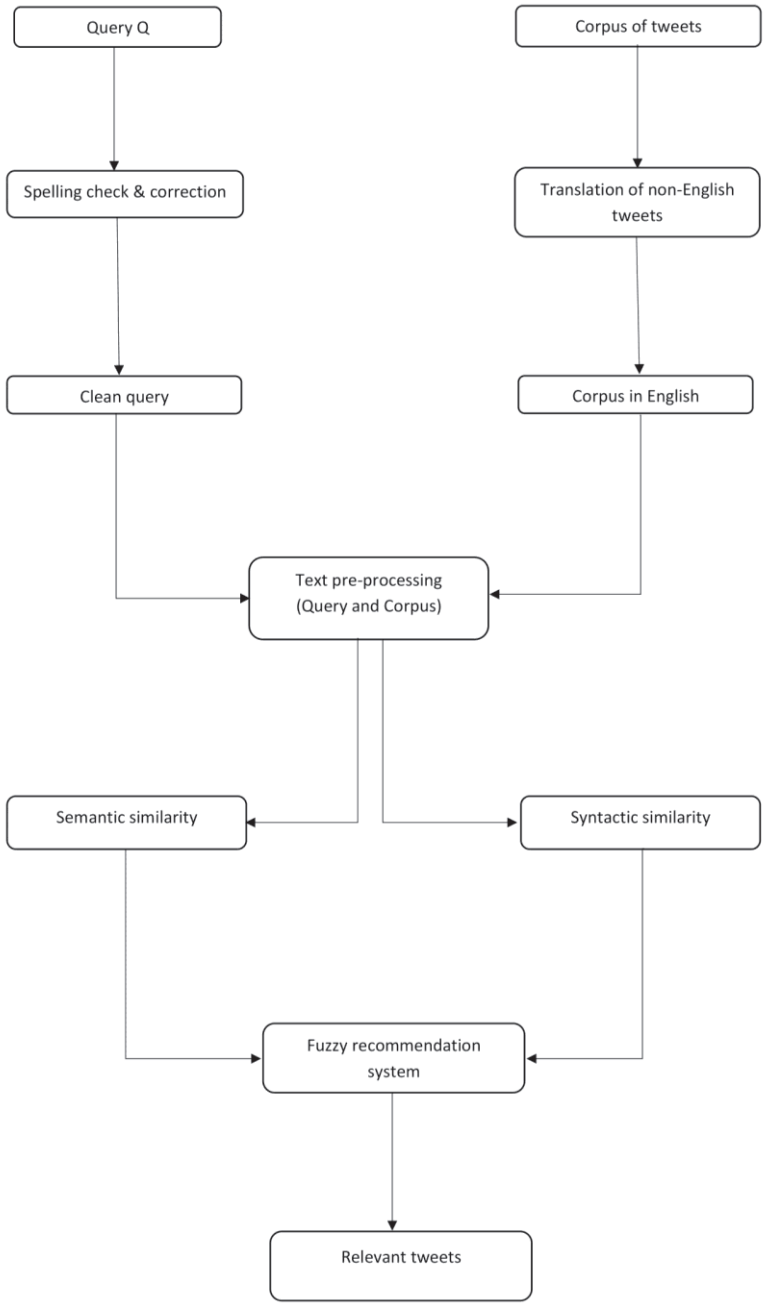

Figure 1. Our proposed system

4.1. Spelling Check & correction

In our everyday writing, there exist different types of errors, one of which that frequently occurs is misspelled a character due to the character's similarity in terms

of sound, shape, and/or meaning. Spelling check is a crucial task to detect and correct human spelling errors in a text. This task is vital for NLP applications such as search engines. Spelling check is a common task in every written language, which is an automatic mechanism to detect and correct human spelling errors. An automatic spelling correction system detects a spelling error and proposes a set of candidates for correction (figure 2) [21]. Researchers divide the whole process into three steps:

1. detection of an error.
2. generation of correction candidates.
3. ranking of candidate corrections

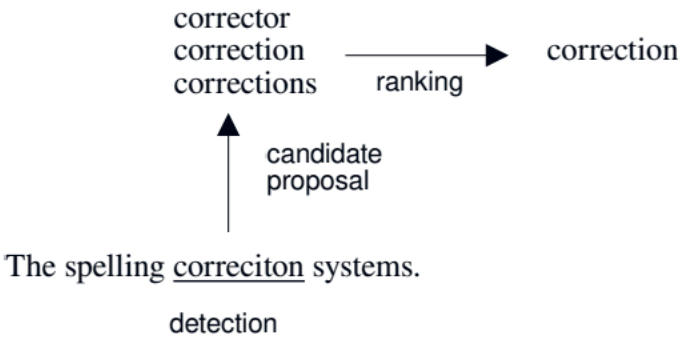

Figure 2. Process of automatic spelling correction

4.2. Translation

Most existing search engines are devoted to query-language entered by the user, while a great share of information is available in other languages. With the growth of the Internet around the world, users can find information in different languages. Searching and getting results in an only single language increases the risks of missing essential information in texts written in other languages.

Our idea is to translate all non-English tweets from the corpus to English, to not have errors in calculating similarity between query and tweets in the corpus, in order to get a good result. For the translation we use a python project library called Translator, this library allows us to translate words or phrases from a language to another one.

To avoid the problems of translation and the errors that can be caused especially, when translating the whole query. We propose to translate it word by word. For that before the translation, we split the tweets into words by removing white spaces, commas, and other symbols, and after all that, we translate it word by word.

4.3. Text pre-processing method

After the step of collecting tweets and translating non-English tweets, we have built our multilingual corpus, however before the use of our methodology, we should make some text pre-processing techniques [22], [23] to prepare the tweets and the query for

the classification and to eliminate the noise existing inside them. in our work we apply some text preprocessing methods such as:

- Tokenization: Which is the phase of splitting the tweet into terms or tokens by removing white spaces, commas, and other symbols etc. This step is very important in our work because we focus on individual words.
- Removing numbers: that do not express any emotions or attitudes. In general, numbers are no use when measuring sentiment and are removed from tweets to refine the tweet content.
- Removing Stopword: There is a kind of word called stopword. They are words of common function in a sentence, such as 'a', 'the', 'to', 'at', etc. These words seem useless for the analysis of the Feeling; therefore, they should be deleted.
- Removing Punctuations: We dont need pits as characteristics, this are only symbols for separate sentences and words, so we delete them from our corpus and the query as well.
- Stemming: Stemming is another very important process. In our work and because we focus on English language, we use the Porter stemming.
- URL and @: The first step is to delete the URL and the word begins with the '@' symbol. We will not follow the content of the Web links, so the URL will be deleted. The '@' symbol has always a username monitoring, which is unnecessary so that the entire word begins with '@' should be deleted.
- Hashtag #: The word begins with '#' is a hashtag. A hashtag is different from other words, it gives a label or a subject on the tweet. Usually, the tag speaks of the subject to which people say in this tweet, and not on the attitudes of the people. This word may provide the information but not important. We have therefore decided not to delete the entire word, but simply delete the symbol '#', and treat the tag as a normal word in a tweet.

4.4. Proposed Fuzzy Logic System

In this subsection, we describe our hybrid approach based on n-grams semantic similarity and reinforcement learning approach, syntactic similarity, and the fuzzy logic system (FLS). As presented before the fuzzy logic system begins with a crisp value and after, fuzzify it, using different steps (fuzzification, rules inference). And finally, return a crisp value in the output using the defuzzification methods (centroid, Mean/, Max...). Figure 3 presents the general structure of a fuzzy logic system.

From figure 3, and as a comparison with our proposed approach, the input(crisp value) of the fuzzy logic system is the two measures (semantic similarity and syntactic similarity) calculated with the n-grams semantic similarity based on reinforcement learning approach [27] and other approach of syntactic similarity, and the output(crisp value) is the class of the tweet (relevant, not relevant), finally return to the user all the relevant tweets sorted by the degree of belonging to the relevant class. As presented earlier, the first step is the definition of the input and the output variables of our proposed FLS. In our case and because we want to classify the tweets according to

two classes (relevant, not relevant), we define two input variables: the semantic similarity and syntactic similarity between the query and the tweets; and one output variable which is the class of the tweet (relevant or not).

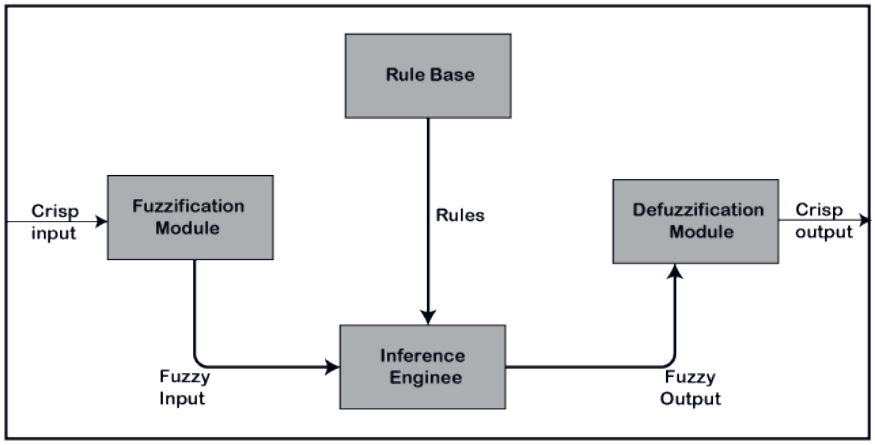

Figure 3. Fuzzy Logic System

In an FLS each variable either in the input or in the output is called linguistic variable, and each linguistic variable has a number of values that can take, these values called linguistic terms or the fuzzy sets. In our case, we have two linguistic variables in the input which are semantic similarity and syntactic similarity, and each one has three linguistic terms which are low, moderate, and high. This means that the semantic similarity score and syntactic similarity score variables can take three possible values, or in other words, can belong to three different fuzzy sets. In the same way, in the output, we have a linguistic variable which is the class of the tweet, and it can also take two different linguistic terms which are relevant, not relevant.

After we have defined the linguistic variables and their linguistic terms in the input and the output the next step of our FLS is the definition of the crisp values of the inputs with which we will begin our approach. For that and as explained in before, we use the n-grams semantic similarity approach based on reinforcement learning to calculate the semantic similarity and syntactic similarity approach of the tweet that will play the role of input's crisp values.

Fuzzification Step: The next step after we calculate the crisp value of each input is the fuzzification step, in which we fuzzify the input variables using the membership function (MF) of each linguistic term. That is, calculating the degree of belonging of the input to each fuzzy set. In this work, we use two different membership functions which are: trapezoidal-shaped MF and the triangular MF [24].

The trapezoidal-shaped MF is a function that depends on four scalar parameters a, b, c, and d, as given by the formula 1 below.

$$f(x) = \begin{cases} 0 & x \leq a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \dfrac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x \end{cases}$$

On the other hand, the triangular MF is a function that depends on three scalar parameters a, b and c, as given by the formula 2 below:

$$f(x) = \begin{cases} 0 & x \leq a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ \dfrac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases}$$

In our case, we need to fuzzify the input variables using one of the MFs presented earlier. For that, we have to define the MF of each linguistic term of the inputs. The linguistic variables "semantic similarity" and "syntactic similarity" have three linguistic terms (three fuzzy sets), so we need to define three MFs, one for the fuzzy set "low", one for "moderate" and another one for "high". The next step for calculating the MFs is the definition of the parameters a, b, c and d for each linguistic term. The choice of these parameters depends on the domain of application of the FLS and needs an expert in this domain. For example, in our case, the values a, b, c, and d will be in the range [0;1], because the inputs "semantic similarity" and "syntactic similarity" have values between 0 and 1 as explained in previous section. The optimal values of these parameters are calculated empirically.

As we presented in section 3, if we want to work without the fuzzy logic concepts, we define for each linguistic term(low, moderate and high) a range between 0 and 1. For example, the semantic similarity and the syntactic similarity is low if they are between 0 and 0.4, moderate if they are between 0.4 and 0.6 and high if they are between 0.6 and 1. So using the classical set each value of the semantic similarity and the syntactic similarity is either belongs to a set(low, moderate or high) with a degree of belonging equal to 1, or not belongs to a set with a degree of belonging equal to 0.

After the membership functions are defined for both input and output, the next step is to define the fuzzy control rules.

Rules Inference: The next step after the fuzzification of the inputs is the step of the definition and the application of the different rules of our problem. That is to say, combine membership functions with the control rules to derive the fuzzy output. We have two inputs (semantic similarity and syntactic similarity) and one output (relevant or not) with three linguistic terms, for that we have defined nine fuzzy rules using the IF-THEN model with the AND logic operation between the values of the inputs. The nine rules of our FLS are the following:

- **IF** Semantic similarity is low **AND** Syntactic similarity is low **THEN** Class is not relevant.
- **IF** Semantic similarity is moderate **AND** Syntactic similarity is moderate **THEN** Class is neutral.
- **IF** Semantic similarity is high **AND** Syntactic similarity is high **THEN** Class is relevant.
- **IF** Semantic similarity is low **AND** Syntactic similarity is moderate **THEN** Class is not relevant.
- **IF** Semantic similarity is low **AND** Syntactic similarity is high **THEN** Class is neutral.
- **IF** Semantic similarity is moderate **AND** Syntactic similarity is high **THEN** Class is relevant.
- **IF** Semantic similarity is moderate **AND** Syntactic similarity is low **THEN** Class is not relevant.
- **IF** Semantic similarity is high **AND** Syntactic similarity is moderate **THEN** Class is relevant.
- **IF** Semantic similarity is high **AND** Syntactic similarity is low **THEN** Class neutral.

After the application of the different rules of our system, the next step is the implication of them to generate the value of each output term. In our case and because we use the AND operation between the inputs, the outputs take the minimum value between them. The last step in the rules inference is the aggregation of the results obtained for each output to find one value for each one.

Defuzzification: After the fuzzification of the inputs and the application of the nine rules of our FLS, we find the degree of belonging of our output (class of the tweet) to each output fuzzy set (relevant, not relevant), and to find the final result of our FLS that have to be in the form of a crisp value, we need to apply the defuzzification step.

The defuzzification process is meant to convert the fuzzy output back to the crisp or classical output to the control objective. The fuzzy conclusion or output is still a linguistic variable, and this linguistic variable needs to be converted to the crisp variable via the defuzzification process. In the literature, there is many defuzzification techniques [16], [25], such as:

- **Max-Membership principle:** This method calculates the maximum between the value of belonging of the output to each fuzzy set.
- **Min-Max Method:** This method calculates the minimum between the value of belonging of the output to each fuzzy set.
- **Mean-Max Method:** This method (also called middle-of-maxima) is closely related to the first method, except that the locations of the maximum membership can be non-unique (i.e., the maximum membership can be a plateau rather than a single point).

In this work, we use the max-membership principle method, to get the maximum between the value of belonging of the output to each fuzzy set. After we find the final crisp value of the output (CVO), the final step is to compare the result obtained with

two range : tweet is not relevant if CVO is between 0 and 0.4, and it is relevant if CVO is between 0.4 and 1.

The final step, after getting the relevant tweets from the fuzzification system, is to sort these tweets by the degree of belonging to the relevant class.

Ranking processing: Following the final phase of the fuzzy logic system, we arrange the tweets that have been identified as relevant based on the relevance score (degree of belonging) obtained during the defuzzification step. Subsequently, we present the top ten tweets associated with the user's query. In this study, we have opted for ten as the threshold for displaying the initial ten results, similar to how search engines operate.

4.5. Example of application

In this subsection, we describe with an example how we classify a query and documents (tweets) in corpus according to two classes (relevant or not relevant) using our proposed approach. For that, we assume that we want to classify a tweet T, so the first step after the translation of T if it was written in another language than English, and the spelling check & correction of the query Q, is the application of the text preprocessing methods. Then we calculate the input variables "semantic similarity" and "syntactic similarity" (crisp values) by the method described earlier using the semantic similarity with the semantic hybrid approach and Cosine similarity. Suppose that after all these steps, we find that the crisp values of the inputs are semantic similarity SM = 0.62 and syntactic similarity ST = 0.22.

After we calculate the crisp values for the inputs, the next step is the fuzzification of these crisp values for that, e use the Trapezoidal-shaped MF presented in formula 3. For all that, we calculate the degree of belonging of the relevant R and the not relevant NR to each fuzzy set (low, high, and moderate) as the following:

- $\mathrm{f}(\mathrm{SM}, \mathrm{low}) = \mathbf{0}, \text{because } \mathrm{SM} = 0.62 \geq \mathrm{d} = 0.35$
- $\mathrm{f}(\mathrm{SM}, \mathrm{moderate}) = \frac{\mathrm{d}-\mathrm{SM}}{\mathrm{d}-\mathrm{c}} = \frac{0.65 - 0.62}{0.65-0.55} = \mathbf{0.3}, \text{because } \mathrm{c} = 0.55 \leq \mathrm{SM} = 0.62 \leq \mathrm{d} = 0.65$
- $\mathrm{f}(\mathrm{SM}, \mathrm{high}) = \mathbf{1}, \text{because } \mathrm{b} = 0.6 \leq \mathrm{SM} = 0.62 \leq \mathrm{c} = 0.8$
- $f(SM, low) = \frac{\mathrm{d}-\mathrm{ST}}{\mathrm{d}-\mathrm{c}} = \frac{0.35 - 0.22}{0.35-0.2} = \mathbf{0.86}, \text{because } \mathrm{c} = 0.2 \leq \mathrm{ST} = 0.22 \leq \mathrm{d} = 0.35$
- $\mathrm{f}(\mathrm{ST}, \mathrm{moderate}) = \mathbf{0}, \text{because } \mathrm{ST} = 0.22 \leq \mathrm{a} = 0.3$
- $\mathrm{f}(\mathrm{ST}, \mathrm{high}) = \mathbf{0}, \text{because } \mathrm{ST} = 0.22 \leq \mathrm{a} = 0.55$

After the fuzzification step, it is the step of the application and the implication of our nine rules as presented below:

- IF (SM is low) = 0 AND (ST is low) = 0.86 THEN (Tweet is Not relevant) = min (0, 0.86) = **0**.

- IF (SM is moderate) = 0.3 AND (ST is moderate) = 0 THEN (Tweet is Neutral) = min (0.3, 0) = **0**.
- IF (SM is high) = 1 AND (ST is high) = 0 THEN (Tweet is relevant) = min (1, 0) = **0**.
- IF (SM is low) = 0 AND (ST is moderate) = 0 THEN (Tweet is not relevant) = min (0, 0) = **0**.
- IF (SM is low) = 0 AND (ST is high) = 0 THEN (Tweet is neutral) = min (0, 0) = **0**.
- IF (SM is moderate) = 0.3 AND (ST is high) = 0 THEN (Tweet is relevant) = min (0.3, 0) = **0**.
- IF (SM is moderate) = 0.3 AND (ST is low) = 0.86 THEN (Tweet is not relevant) = min (0.3, 0.86) = **0.3**.
- IF (SM is high) = 1 AND (ST is moderate) = 0 THEN (Tweet is relevant) = min (1, 0) = **0**.
- IF (SM is high) = 1 AND (ST is low) = 0.86 THEN (Tweet is neutral) = min (1, 0.86) = **0.86**.

The next step is the aggregation of these rules for each output fuzzy set.

- Tweet is neutral → max (0, 0, 0.86) = 0.86
- Tweet is not relevant → max (0, 0, 0.3) = 0.3
- Tweet is relevant → max (0, 0, 0) = 0

So, after all these steps, we find the degree of belonging of the output to each output fuzzy set, and by using the MFs of the output and the method of Max-Membership we defuzzify the output to find the final crisp value of the output (class of the tweet).

After the defuzzification step using the Max-Membership method, the final crisp value obtained by applying our approach is equal to 0.6, and because this value is between 0.6 and 1, the tweet T is relevant to the user's query.

5. Experimental Results

In this section, we will present some experimental results of our work. As shown earlier, the first step is to create a tweet dataset using the Twitter API. For that, our dataset contains the tweets related to covid pandemic. that is, we will classify tweets related to this subject.

As presented above, our Fuzzy Logic System contains two input variables : similarity syntactic and semantic similarity based on n-grams and reinforcement learning and an output variable (class of the tweet : relevant or not), the range of each variable is between 0 and 1 (because all the similarities scores are between 0 and 1), and each has three linguistic terms (linguistic values or fuzzy sets): low, moderate and high for the input variables, and relevant and not relevant for the output variable.

In our FLS we use two membership functions for the fuzzification that are: Trapezoidal MF and Triangular MF, nine IF-THEN rules and four defuzzification

methods (Max-Membership, Min-Max, Mean-Max). From that, to make a choice of the best membership function and the best method of defuzzification for our system, we have six possible combinations for making this choice. The table 2 and the figure 4 show respectively the results obtained for the error rate and the classification rate after the classification of the tweets using six possible combinations: Trapezoidal MF/Min-max, Trapezoidal MF/Mean-max, Trapezoidal MF/Max-Membership, Triangular MF/Min-max, Triangular MF/Mean-max, Triangular MF/Max-Membership.

Fuzzification	Defuzzification	ER	Accuracy
Trapezoidal MF	Min-Max	69%	31%
	Mean-Max	42%	58%
	Max-Membership	7%	93%
Triangular MF	Min-Max	69%	31%
	Mean-Max	68%	32%
	Max-Membership	17%	83%

Table 2. Error rate and Accuracy using Fuzzification/Defuzzification combinations.

From the table 2, the best fuzzification/defuzzification combination is the one which we use the Trapezoidal MF for the fuzzification step and Max-membership method for defuzzification, with an accuracy of 93% and error rate of 7%. From these results we note the use of this combination increase the classification rate in our system compared to other combinations, for that we decide to use in our FLS the Trapezoidal MF for the fuzzification of our two input (semantic and syntactic similarity) and Max-membership method for defuzzification step to find the class of the tweet (relevant or not).

The next experiment is to compare our approach based on fuzzy logic (A) and other approaches from the literature such as: Logistic Regression (B), Gaussian Naive Bayes (C) and Decision Tree (D). For that we compute the accuracy, error rate, recall and F1 score of the classification of the tweets using all the approaches cited above, and compare them with the results obtained with our approach based on semantic/syntactic similarity using fuzzy logic.

The figure 4 shows the results obtained.

According to Figure 4, our approach (A) based on fuzzy logic, syntactic similarity, and semantic similarity using reinforcement learning to classify the tweets in our dataset and return the most relevant tweet outperforms other approaches. Our approach achieved a classification rate of 93%, with an error rate of only 7% when compared to existing literature, such as [12] which employs Decision Tree to enhance classification rates, and Shah et al in [13] who present a text classification model based on the Logistic Regression algorithm to improve classification results. We also compared our approach with the one presented in [14], which relies on Naive Bayes classifiers, and their studies show that Gaussian Naive Bayes improves classification rates. It's worth noting that none of these previously mentioned approaches from the literature used fuzzy logic in their text classification.

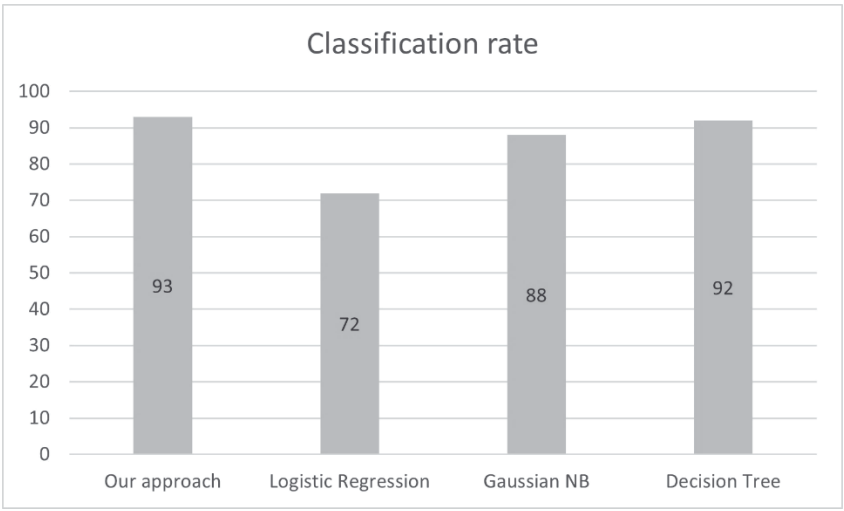

Figure 4. Classification Rate

However, based on the data from Figure 4, we conclude that our proposed approach has significantly improved the results in the classification of textual documents. This clearly demonstrates the efficacy of fuzzy logic in enhancing classification quality by increasing the classification rate while simultaneously reducing the error rate.

In the next experiment, we decide to calculate some other metrics to approve that our approach improves the classification rate compared to other approaches from the literature, for that we will calculate the precision, recall and F1-score of our approach, and compare them with the same metrics of the 3 other approaches that are cited in the previous experiment.

	Precision	**Recall**	**F1-Score**
Our approach (A)	97%	93%	95%
Logistic Regression (B)	72%	99%	83%
Gaussian NB (C)	94%	89%	91%
Decision Tree (D)	99%	89%	94%

Table 3. Comparison of metrics between different approaches.

From the table 3 and figure 5 we note that our proposed approach A based on fuzzy logic gives the good result either for the accuracy, precision, recall, and F1-score in comparison with the other approaches. It has a good accuracy 93% and it improves the F1-score as 95%. All that demonstrates how our proposed approach outperforms the other approaches.

The idea of this work is coming from this paper [27], for that, the last experiment is to compare the result of our new approach based on fuzzy logic to our last approach of semantic similarity based on reinforcement learning (RL), and the syntactic similarity approach: cosine similarity.

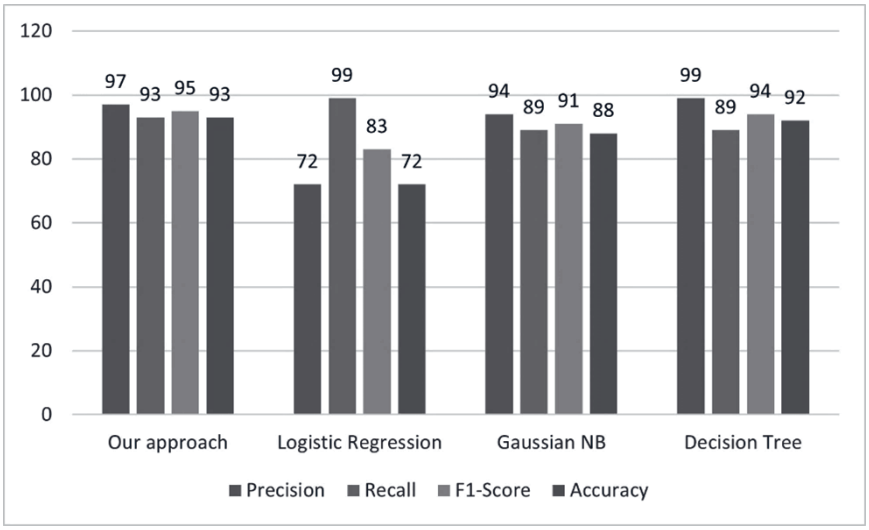

Figure 5. Evaluation of our system

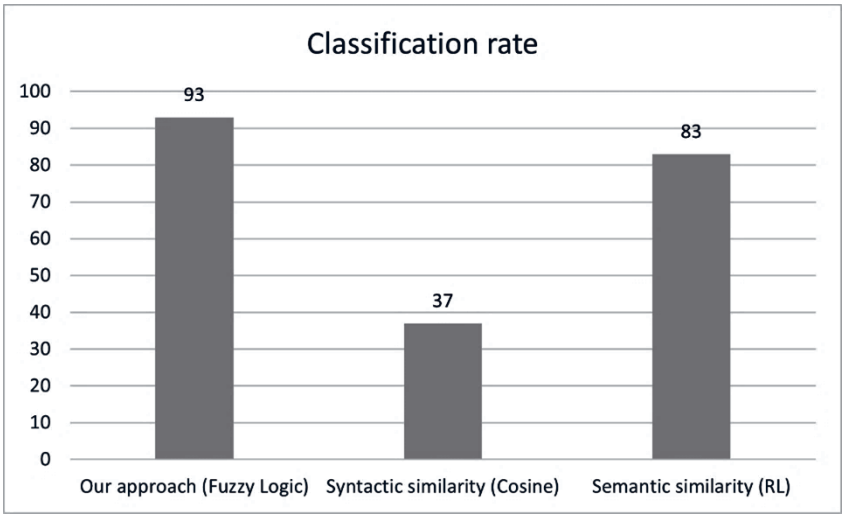

Figure 6. Comparison of our approach with similarity approaches

According to figure 6, we remark that with our new approach based on fuzzy logic, we increase the percentage of the classification rate (from 83% to 93%) and decreases that of the error rate (from 17% to 7%) comparing with our last approach that uses only the semantic similarity without fuzzy logic. We notice from the figure that our approach based on fuzzy logic gives better results compared to the syntactic similarity approach too. Therefore, we can say that by adding the fuzzy logic we can improve the classification rate, then the quality of our results and system.

6. Conclusion

In this article we have presented a new approach that search and return the most relevant result to a query entered by a user of a search engine, our new approach is based on fuzzy logic and use two type of similarity the first one is syntactic similarity which is cosine similarity, and the second one is a semantic similarity based on reinforcement learning and n-grams model. The experimental results show that using fuzzy logic approach gives us better results, and that our proposed approach outperforms some other methods from the literature.

Compliance with ethical standards
Funding declaration: The authors declare that they have no funding for this work.
Authors's contributions: In this article we present a new fuzzy recommendation system based on syntactic and similarity measures and n-grams model, to get the most relevant documents for a user in a search engine. All authors read and approved the final manuscript.
Competing interest: The authors declare that they have no conflict of interest.

References

[1] R. Kumar and S. C. Sharma, "Information retrieval system: An overview, issues, and challenges," International Journal of Technology Diffusion (IJTD), vol. 9, no. 1, pp. 1-10, 2018.

[2] Y. Djenouri, A. Belhadi, P. Fournier-Viger, and J. Lin, "Fast and effective cluster-based information retrieval using frequent closed itemsets," Information Sciences, vol. 453, pp. 154-167, 2018.

[3] G. H. Nguyen, L. Tamine, L. Soulier, and N. Souf, "A Tri-Partite Neural Document Language Model for Semantic Information Retrieval," in Proc. of the 15th European Semantic Web Conference, ESWC 2018, Springer, pp. 445–461, 2018.

[4] S. Kumar and K. K. Bhatia, "Semantic similarity and text summarization-based novelty detection," SN Applied Sciences, 2020.

[5] J. Martinez-Gil, R. Mokadem, J. Küng, and A. Hameurlain, "A Novel Neurofuzzy Approach for Semantic Similarity Measurement," in International Conference on Big Data Analytics and Knowledge Discovery, Springer, pp. 192-203, 2021.

[6] V. Cross and V. Mokrenko, "Fuzzy Set Similarity Between Fuzzy Words," in International Fuzzy Systems Association World Congress, Springer, pp. 214-223, 2019.

[7] D. V. Suma, "A novel information retrieval system for distributed cloud using hybrid deep fuzzy hashing algorithm," Journal of Information Technology and Digital World, vol. 2, no. 3, pp. 151-160, 2020.

[8] S. H. Farhi and D. Boughaci, "Graph based model for information retrieval using a stochastic local search," Pattern Recognition Letters, vol. 105, pp. 234-239, 2018.

[9] D. P. Joby, "Expedient information retrieval system for web pages using natural language modeling," Journal of Artificial Intelligence and Capsule Networks, vol. 2, no. 2, pp. 100-110, 2020.

[10] B. Selvalakshmi and M. Subramaniam, "Intelligent ontology based semantic information retrieval using feature selection and classification," Cluster Computing, vol. 22, pp. 12871-12881, 2019.

[11] M. Esposito, E. Damiano, A. Minutolo, G. De Pietro, and H. Fujita, "Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering," Information Sciences, vol. 514, pp. 88-105, 2020.

[12] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 20-28, 2021.

[13] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and KNN models for text classification," Augmented Human Research, vol. 5, pp. 1-16, 2020.

[14] V. Venkatesh, K. V. Ranjitha, and B. S. Venkatesh Prasad, "Optimization scheme for text classification using machine learning na¨ıve bayes classifier," in ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications, Springer Singapore, pp. 576-586, 2020.

[15] L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8, no. 3, pp. 338-353, 1965.

[16] S. Chakraverty et al., "Defuzzification," in Concepts of Soft Computing: Fuzzy and ANN with Programming, pp. 117-127, 2019.

[17] S. Thaker and V. Nagori, "Analysis of fuzzification process in fuzzy expert system," Procedia computer science, vol. 132, pp. 1308-1316, 2018.

[18] L. A. Zadeh and R. A. Aliev, Fuzzy logic theory and applications: part I and part II, World Scientific Publishing, 2018.

[19] H. T. Nguyen, C. L. Walker, and E. A. Walker, A first course in fuzzy logic, CRC press, 2018.

[20] T. Rattanasawad et al., "A comparative study of rule-based inference engines for the semantic web," IEICE TRANSACTIONS on Information and Systems, vol. 101, no. 1, pp. 82-89, 2018.

[21] D. Hládek, J. Staš, and M. Pleva, "Survey of automatic spelling correction" Electronics, vol. 9, no. 10, pp. 1670, 2020.

[22] A. I. Kadhim, "An evaluation of preprocessing techniques for text classification," International Journal of Computer Science and Information Security (IJCSIS), vol. 16, no. 6, pp. 22-32, 2018.

[23] L. Hickman et al., "Text preprocessing for text mining in organizational research: Review and recommendations," Organizational Research Methods, vol. 25, no. 1, pp. 114-146, 2022.

[24] A. Jain et al., "Membership function formulation methods for fuzzy logic systems: A comprehensive review," Journal of Critical Reviews, vol. 7, no. 19, pp. 8717-8733, 2020.

[25] K. S. Gilda and S. L. Satarkar, "Analytical overview of defuzzification methods," International Journal of Advance Research, Ideas and Innovations in Technology, vol. 6, no. 2, pp. 359-365, 2020.

[26] S. Jain, K. R. Seeja, and R. Jindal, "A fuzzy ontology framework in information retrieval using semantic query expansion," International Journal of Information Management Data Insights, vol. 1, no. 1, p. 100009, 2021.

[27] A. El Hadi et al., "Finding Relevant Documents in a Search Engine Using N-Grams Model and Reinforcement Learning," Journal of Information Technology Research (JITR), vol. 15, no. 1, pp. 1-17, 2022.

[28] L. A. Zadeh, "Fuzzy logic—a personal perspective," Fuzzy sets and systems, vol. 281, pp. 4-20, 2015.

[29] H. Liu and M. Cocea, "Fuzzy rule-based systems for interpretable sentiment analysis," in 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), pp. 129-136, IEEE, 2017.

[30] B. R. Boyce et al., Text information retrieval systems, Elsevier, 2017.

[31] D. Sharma, R. Shukla, A. K. Giri, and S. Kumar, "A brief review on search engine optimization," in 2019 9th international conference on cloud computing, data science & engineering (confluence), pp. 687-692, IEEE, 2019.