

A novel fuzzy logic-based approach for textual documents indexing

Latifa Rassam¹, Imane Ettahiri², Ahmed Zellou¹, Karim Doumi²

¹Software Project Management Research Team, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Mohammed V University, Rabat, Morocco

²Alqualsadi Research Team, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Mohammed V University, Rabat, Morocco

Article Info

Article history:

Received Dec 3, 2023

Revised Jan 6, 2024

Accepted Jan 11, 2024

Keywords:

Fuzzy indexing

Fuzzy logic

Keywords extraction

Natural language processing

N-gram graph

ABSTRACT

In the evolving landscape of information retrieval and natural language processing, the quest for more effective automatic keyword extraction (AKE) techniques from textual documents has become a pivotal research focus. Existing methodologies, while offering valuable insights, often grapple with the challenges posed by the imprecision and variability inherent in human language. This has led to a growing recognition of the need for innovative approaches to navigating textual content's nuances more adeptly. In response to this imperative, this paper proposes a novel fuzzy indexing approach designed specifically for the indexing of textual documents. Fuzzy indexing, grounded in the principles of fuzzy logic, provides solutions for handling the inherent uncertainty and imprecision in natural language, especially when confronted with the intricacies of linguistic ambiguity and variability. By leveraging the power of fuzzy logic, we aim to enhance the precision of keyword extraction. This paper unfolds the intricacies of our fuzzy indexing approach, detailing the theoretical methodology through empirical evaluation and comparative analysis; we seek to demonstrate the efficacy of our approach in outperforming traditional methods in the context of fuzzy indexing for textual documents.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Latifa Rassam

Software Project Management Research Team, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Mohammed V University

Avenue Mohammed Ben Abdellah Regragui, Madinat Al Irfane, BP 713, Agdal Rabat, Morocco

Email: rassamlatifa@gmail.com

1. INTRODUCTION

The classical form of logic plays a significant role in various disciplines and has a wide-ranging application [1]. However, it is limited to expressing data with Boolean values, namely true and false. To address this limitation, fuzzy logic was introduced to allow for the characterization of elements in a gradual manner [2]. In this context, Professor Zadeh proposed fuzzy logic as an extension of Boolean logic in the 1960s [3]. In this field, precision and strictness in statements are not the primary focus; instead, the emphasis lies on the ability to handle ambiguous propositions [4]. This manuscript specifically aims to develop a robust quantification system for measuring the degree of relevance, known as connection, focusing on achieving a higher level of comprehension accuracy for key expressions. Besides, indexing plays a crucial role in the data processing sequence, as it involves searching to determine the most effective approach to extract the essential information that fulfills the requirements [5]. The data retrieval process should be swift and efficient, highlighting the significance of using the textual document as a foundational structure [6].

Previous studies [7]–[9] have introduced unsupervised keyword extraction approaches that incorporate n-grams with word and document embedding [10]. These approaches include methods for composing document vectors using word vectors and their idf-scores [11], as well as leveraging higher-order word n-grams to enhance unigram embedding and generate document embedding [12]. Performance evaluations on multiple datasets demonstrated the effectiveness of combining higher-order word n-grams, retrofitted glove embedding [10], and document embedding for keyword extraction. Notably, the use of bi-gram retrofitted embedding [13] led to significant improvements compared to baseline approaches.

Li *et al.* [14] introduce a probabilistic model that operates in a semi-supervised setup. Their approach integrates graph-based information from a document into a Bayesian framework by utilizing an informative prior. This incorporation enables their model to support formal statistical inference, enhancing its effectiveness in capturing document characteristics. They used a fuzzy inference system to calculate sentence scores and applied bidirectional gated recurrent units to remove redundant or similar sentences. Additionally, they generated abstractive summaries based on the selected sentences [15].

Terrada *et al.* [16], the authors propose a post-processing approach to enhance the performance of automatic keyword extraction (AKE) methods by incorporating semantic awareness through part-of-speech (PoS) tagging. Conneau *et al.* [17] evaluate their supervised approach by considering word types obtained from PoS tagging, specialized terms from context-dependent thesauri, and named entities [16]–[18] as sources of semantic information. The authors demonstrate the positive impact of their approach on improving AKE methods by integrating these semantic elements. Ye *et al.* [19] and Tsukagoshi *et al.* [20] focused on improving the processing of long multi-documents.

These indexing approaches face three main challenges. Firstly, they often produce multiple potential indexes, including both correct and incorrect ones. As a result, human intervention is required to determine the accuracy of the generated indexes. Secondly, many of these techniques are designed for structured corpora, neglecting the potential benefits of indexing large unstructured documents. The structure of textual documents, including subtitles, keywords, titles, and chapters, contains valuable information that is often overlooked. Finally, existing indexing approaches and performance evaluation techniques exhibit limited accuracy when it comes to matching indexes with annotated keywords in textual documents.

Therefore, the most important endowment of our work involves mainly these four contributions:

- We are proposing a new optimal n-gram graph-based technique for textual document indexing.
- We are proposing an improved algorithm for keyword extraction.
- Coming up with new fuzzy logic-based techniques for calculating the degree of relevance of the generated keywords within a corpus of textual documents.
- We implement and evaluate the proposed technique on a real-world domain dataset with annotated keywords.

Comparing our proposed technique to other models shows that our approach can reach a high level of accuracy using some important measures (precision, recall, and overall) and empirical results indicate that our technique outperforms plenty of models in terms of precision and accuracy. The subsequent sections of this paper are structured as follows: In section 1, we delve into the introduction, motivation, and contributions of this research. In section 2 elucidates the fuzzy logic-based method of n-gram word indexing for textual document representation. Section 3 presents the experimental results and provides a comprehensive analysis and interpretation. Finally, in section 4, we offer concluding remarks on this study and outline potential avenues for future research exploration. In the next section, we will present our fuzzy logic-based approach, which is a solution to the performance of index generation problems such as allowing the representation of uncertainty and imprecision in linguistic expressions, enabling a more flexible and nuanced handling of document content.

2. METHOD

2.1. Using N-gram graph process for indexing

The importance of a specific document is directly related to the meaning and significance of each sentence and the corresponding terms within them, which collectively contribute to the knowledge conveyed in the document [21]. Therefore, to quantify the significance of the document, it becomes necessary to quantify the importance of each term within it. The schema in Figure 1 describes decently our indexing approach process from the split phase to the matching accuracy phase.

The process of fuzzy indexing involves calculating the membership degree of each term in the document's fuzzy set after transforming the textual document into an n-gram graph, then considering each n-gram word as a fuzzy term. This is done by considering the linguistic characteristics and context of the terms within the document and comparing them to a predefined fuzzy vocabulary or knowledge base. The resulting fuzzy representation of the document enables more sophisticated information retrieval techniques, including

fuzzy matching, fuzzy clustering, and fuzzy similarity comparisons, to be applied in various applications such as text summarization, information retrieval, and content-based document comparison.

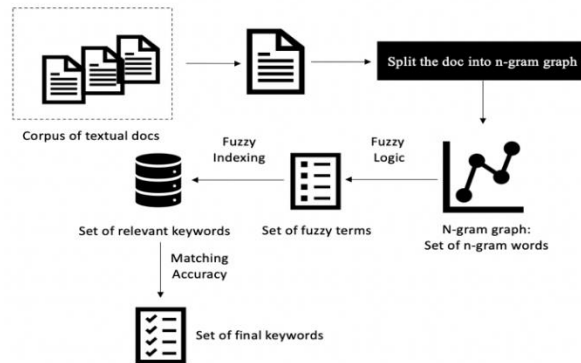


Figure 1. Our fuzzy indexing approach process

Consider the following definitions: Let: $S = \{S_1, S_2, \dots\}$ represent the set of vertices in the graph, where each S_i corresponds to a keyword extracted from the textual document td . Then $\Omega = \{V_\Omega, E_\Omega, L, W\}$ represents the graph, where:

- $V_\Omega = S_i$ is the set of vertices v in the graph.
- E_Ω Is the set of edges, denoted as e , with each edge having the form $e = \{v_1, v_2\}$, indicating a connection between vertex v_1 and vertex v_2 .
- $L: V_\Omega \rightarrow B$ is a function that assigns labels to the set of vertices. It assigns a label to each vertex in the graph.
- $W: E_\Omega \rightarrow Q$ is a function that assigns weights, denoted as $w(e)$, to the set of edges. It assigns a weight value to each edge in the graph.

Figure 2 describes decently the first phase of our indexing process from a structured textual document to an N-gran graph; starting with the preprocessing step, then the transformation step namely the tokenization using our improved Levenshtein algorithm see in algorithm 1:

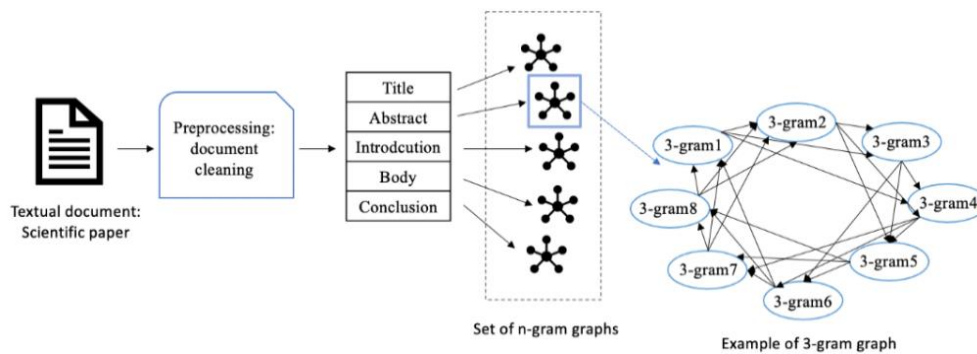


Figure 2. Our detailed word extraction process using the N-gram graph approach

We have optimized the algorithm that we have proposed to resolve the limitations that face a lot of other algorithms such as the Jaccard similarity algorithm [22] and cosine similarity algorithm [22] by using firstly, the porter stemming function [23] reduces words to their root forms, which helps consolidate similar words and reduce vocabulary size. Secondly, a set of stop words from the natural language toolkit (NLTK) library [24] to filter out n-grams that contain commonly occurring and less informative words. Thirdly, the NLTK library’s N-grams function to efficiently generate N-grams from the document. Finally, the defaultdict class from the collection module is used to store the n-gram graph, providing a convenient way to manage occurrence counts [25].

Algorithm 1. The tokenization from textual document to N-gram graph algorithm

```

Start
The inputs:
    ptd: A specific preprocessed textual document
    n: the number of n-grams
The outputs: the set of n-gram word  $S\Omega_m$ 
 $S\Omega_m \leftarrow \text{defaultdict}(\text{int})$ 
    stemmer  $\leftarrow$  PorterStemmer()
    stop_words  $\leftarrow$  set(stopwords.words('English'))
    For ngram in ngrams (ptd, n):
        stemmed_ngram  $\leftarrow$  tuple(stemmer.stem(token))
        If any (token in stop_words for token in stemmed_ngram):
            ngram_str  $\leftarrow$  join((stemmed_ngram), ' ')
             $S\Omega_m[\text{ngram\_str}] + 1$ 
        End if
    End for
Return  $S\Omega_m$ 
End
    
```

2.2. Quantifying N-gram word occurrence using our fuzzy logic approach

In the context of fuzzy logic, a document, which is transformed into an n-gram graph, is represented as a set of fuzzy terms denoted as x. To determine the value and significance of a specific expression within a textual document, it is imperative to calculate the membership degree of that expression within the fuzzy set. The relationship between the document and the expression can be defined as (1).

$$Dp = TD \times \mathbb{T} \rightarrow [0,1], Dp(td_x) = Degree(t, td) \tag{1}$$

Where:

\mathbb{T} : is the set of all terms

TD : is the set of documents within the corpus.

The mentioned function (1) assigns a weight value to each term index, which plays a crucial role in determining the relevance of a document. The search aspect primarily focuses on how the weight calculation function computes term degrees. Various functions exist that calculate term frequencies within a specific document or a large corpus of textual documents [26]. A keyword is considered highly valuable if it appears in the document’s title or abstract [27]. Conversely, if it appears in the footnotes or references section, its importance diminishes. Consequently, the repetition score of a term can be easily determined by examining its location within the document.

We have come up with a fuzzy description for textual documents, which facilitates the quantification of word importance. To determine the degree of importance of a word, represented by a specific term (t), within a textual document (td), we calculate the ratio of its importance within each constituent part (m) of the document. Each section carries a certain level of importance, denoted by values between 0 and 1 [28], [29], indicating where the search should focus for related terms. Subsequently, for each section, we require a mathematical function that quantifies the significance level of the term (t) in that particular section (S_i). Finally, the various degrees associated with different parts are typically linked to a fuzzy function that encompasses multiple linguistic quantifiers [30]. These quantifiers, such as “all,” “at least once upon a time,” “about multiple different times,” and others, generally reflect a user’s preference regarding the presence requirement of a term (typically a word) [31]. To quantify the membership degree of a particular expression x_i , we employ our core mathematical function called *Calculatoccurence*(t,td). This function calculates the count of occurrences of the expression x within the specific textual document td. By applying this function, we can determine the degree to which expression x_i is present or relevant within td_1 [32]:

$$Degree(x_i, td) = \frac{\sum_{i=1}^n \text{Calculatoccurence}(x_i, td)}{m} \tag{2}$$

Here, m refers to the occurrence number of the term that is repeated the most of time within the textual document td. It serves as a normalization factor to determine the relative relevance of other terms in comparison to the most frequently repeated term [33]. To calculate the relevance degree of a term within a textual document, we have proposed (1) and (2) which quantify the degree of relationship of an expression within a specific document. Here, we can distinguish two main cases: to assess the degree of relevance between two terms within the same textual document (namely two N-gram words in the same graph), we propose the function (3); to assess the degree of relevance between two terms in two entirely different textual documents (namely two N-gram words in two different graphs), we propose and employ the calculation method in (4).

This implementation not only highlights the practicality of the approach but also emphasizes its potential for enhancing overall performance in handling textual data. We conducted these evaluations using extracts from our datasets. The results demonstrated a notable improvement in efficiency, highlighting the effectiveness of the proposed fuzzy functions (5) and (6). The ‘n’ represents the count of grams used in the indexing procedure of the document in question; it refers to the same n in the n-gram graph concept. Therefore, the general representation of our proposed fuzzy functions will be as represented in (7) for the case of calculating the fuzzy precision value of different terms in the same textual document, in (8) for the other case of calculating the fuzzy precision value of multiple different terms in many distinct textual documents.

$$\varphi x_{i,j}(td_1) = \frac{\text{Maximum}((\varphi x_i(td_1)),(\varphi x_j(td_1))) \times n}{n-1} \quad (3)$$

$$\psi x_{i,j}(td_1, td_2) = \frac{\text{Minimum}((\varphi x_i(td_1)),(\varphi x_j(td_2))) \times n}{n-1} \quad (4)$$

$$\varphi x_{i,j,w}(td_1) = \frac{\text{Maximum}((\varphi x_i(td_1)),(\varphi x_j(td_1)),(\varphi x_w(td_1))) \times n}{n-1} \quad (5)$$

$$\psi x_{i,j,w}(td_1, td_w) = \frac{\text{Minimum}((\varphi x_i(td_1)), \dots, (\varphi x_w(td_w))) \times n}{n-1} \quad (6)$$

$$\varphi x_{i,j}(td_1) = \begin{cases} 0 \\ \frac{\text{Maximum}((\varphi x_i(td_1)),(\varphi x_j(td_1))) \times n}{n-1} \\ 1 \end{cases} \quad (7)$$

$$\psi x_{i,j}(td_1, td_2) = \begin{cases} 0 \\ \frac{\text{Minimum}((\varphi x_i(td_1)), \dots, (\varphi x_w(td_w))) \times n}{n-1} \\ 1 \end{cases} \quad (8)$$

The fuzzy value, in (7) and (8), of 1 indicates that the generated keywords have been found in all the textual documents within the corpus under consideration. This implies that the keywords are present across the entire corpus. A fuzzy value between ‘0’ and ‘1’ suggests that the index or keyword has appeared multiple times in different textual documents within the corpus. While not present in every document, it demonstrates a recurring presence throughout the corpus. Finally, a fuzzy value of 0 signifies that the keyword in question is not present in any of the textual documents within the corpus. This means that none of the documents contains the specific keyword.

3. EVALUATION

3.1. Datasets

In this section, we will introduce the datasets employed to assess our fuzzy logic-based approach. Furthermore, we will determine the superior one among the four approaches used for AKE by evaluating their performance through precision, recall, and overall metrics. Subsequently, we will conduct a comparative analysis of our method’s performance against other AKE methods.

Table 1 outlines the datasets employed by the methods under investigation. Notably, methods leveraging deep learning (DL) techniques necessitate extensive datasets for effective training. Unfortunately, up until 2017, the largest available database contained merely 2,304 scientific articles [34], rendering it insufficient for training recurrent neural networks (RNN). However, the introduction of KP20K, comprising 527,430 documents, has become the preferred dataset for methods emerging after 2017. Furthermore, the majority of the methods under examination rely on three datasets to evaluate their performance; these are detailed in Table 1 as performance evaluation datasets for AKE methods. It is noteworthy that a recent dataset, KPTime [35], consisting of 259,923 training documents, has been introduced but remains untapped in the evaluation of AKE methods.

Table 1. The datasets

Dataset	Documents	Documents type	Language	Tokens	Annotation	Usage rate
NUS [36]	211	Full paper	English	8398.30	Reader	50%
Krapivin [34]	2304	Full paper	English	8040.74	Author	42%
Semeval [36]	244	Full paper	English	7961.20	Both	58%

3.2. Evaluation metrics

In addition to our fuzzy logic-based degree of relevance calculation approach, we employ the following metrics [37] to evaluate the accuracy of our generated indexes using our new approach. The results of these measurements remain relative because the number of words extracted and the nature and length of the document affect them. In addition, methods that do not predict phrases that do not exist in the document will have fewer results when using these measures. It will therefore be necessary to think about other ways of evaluating performance that go beyond these constraints. These measures serve as key indicators of index quality (9).

$$Precision = \frac{Accurate\ index}{Accurate\ index + Inaccurate\ index} \tag{9}$$

As shown in (9) this measure assesses the accuracy of the generated indexes.

$$Recall = \frac{Accurate\ index}{Missed\ index + Accurate\ index} \tag{10}$$

As shown in (10) measures the effectiveness of the generated indexes by evaluating the proportion of relevant terms that are correctly identified and included in the index.

$$Overall = Recall \times \left(2 - \frac{1}{Macro\ Precision} \right) \tag{11}$$

As shown in (11) evaluate and validate the accuracy and effectiveness of our new approach in generating indexes.

3.3. Results and discussions

When applying an n-gram graph-based process with a measurement of 3 to the initial part of our textual document, it yields 27 N-gram words. On the other hand, when using a measurement of 4 for the same initial part, the N-gram graph-based process returns 26 N-gram words. Table 2 describes the number of keywords and the set of indexes that were generated by indexing the first paragraph (title) of the concerned textual document using the n-gram graph approach in the case n = (2, 3, 4), which means after splitting the paragraphs into words that contain respectively 2, 3, and 4 words.

From Table 3, it is evident that the key indicator membership degree shows an increase when using word graph-based indexing with a specific number of N-grams (n=2). Conversely, this indicator decreases when using word indexing with n=4, while its value is average for n=3. This implies that the graph-based word indexing technique with n=2 is our case's most convincing approach for textual document indexing. The graph in Figure 2 illustrates the output, which is a 3-gram graph example after textual document tokenization after splitting the first section of our textual document into 3-word grams.

Each of the graphs will consist of 14 vertices for the corresponding values of n, which are one, two, and three as represented in Tables 2 and 3. To comprehensively evaluate the quality of the indexing performed and the generated indexes, it was crucial to consider additional quality factors and measures beyond the number of generated N-grams, relevance degree, and membership degree for each index. These factors include macro precision, recall, and overall performance which heavily rely on the number N.

Table 2. The number of extracted keywords per N value

N Value	2	3	4
Index	The 1st set of indexes	The 2nd set of indexes	The 3rd set of indexes
Number of keywords	28	27	26

Table 3. The keywords membership degree calculation depends on the n value

Index	The 1st set of indexes	The 2nd set of indexes	The 3rd set of indexes
Membership degree for n = 1	0.65	0.75	0.8
Membership degree for n = 2	0.54	0.6	0.72
Membership degree for n = 3	0.48	0.55	0.62
Membership degree for n = 4	0.45	0.50	0.57

By analyzing these values, we can understand the nature of the relationship between the various factors and assess the overall quality of the indexing process and the resulting indexes. The graph displayed in Figure 3 illustrates the variation in the value of the macro precision measure used during our analysis, depending on the value of N (the n contained in N-gram) compared to the precision value of the other extraction

systems. The value of metrics for the three others approaches have been calculated using the fuzzy function that we proposed and based on the results provided by their empirical studies.

Figure 4 describes the recall value of our novel fuzzy indexing approach compared to the other techniques. It indicates that by using a smaller n value (such as 2) the recall value is increased to 0.66 while the recall value is reduced for the other indexing techniques; this is a significant improvement that means a high level of accuracy compared to the other techniques. This large increase is not unexpected considering the fact that the tokenization algorithm was designed based on several mathematical optimizations. The single most surprising result can be seen in Figure 5 is the overall value of n=2 compared to the other techniques which indicates a higher growth and improvement.

In an ideal scenario, all metrics would reach their maximum value of 1, indicating a perfect combination. Therefore, the perfect scenario occurs when the following conditions are met: macro precision = overall = recall = 1, which indicates optimal performance across all metrics. We proceeded to evaluate and compare the indexes generated by our fuzzy logic-based approach on a real-world domain dataset with previously published results from three extraction systems that were previously mentioned in the chapter of related work (enhanced word and document embedding [11], PoS-Tagging and enhanced semantic-awareness [17], fuzzy Bi-GRU hybrid approach for extractive, and abstractive summarization of long multi-documents [15]).

The main evaluations are based on three main components macro precision, recall, and overall are used to evaluate alignments between classes and properties. We evaluated our fuzzy logic-based approach using the same real-world domain dataset. We observed a high level of matching accuracy achieved, as indicated by the average values of the main metrics (for N=2) defined on three reference alignments: macro precision = 0.89, recall = 0.66, and overall = 0.57. In comparison, the state-of-the-art indexing systems exhibited a lower level of matching accuracy (about our approach), with average metrics falling within the ranges: $0.4 \leq \text{macro precision} \leq 0.7$, $0.2 \leq \text{recall} \leq 0.65$, and $-0.2 \leq \text{overall} \leq 0.34$.

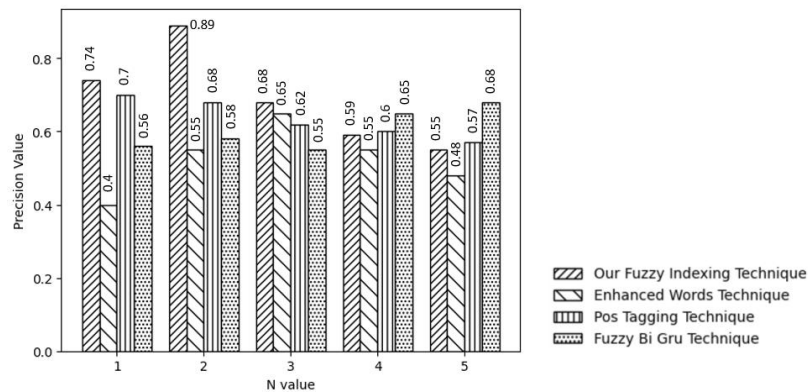


Figure 3. Our fuzzy indexing technique's macro precision value representation compared to other indexing techniques per N value

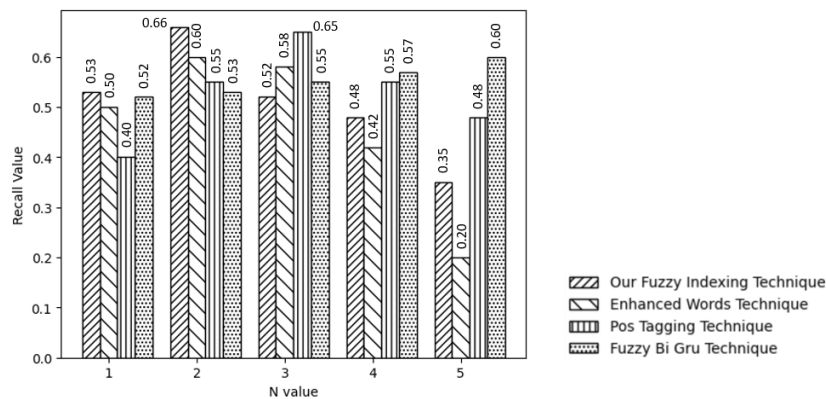


Figure 4. The recall value representation of our fuzzy indexing technique compared to other indexing techniques per N value

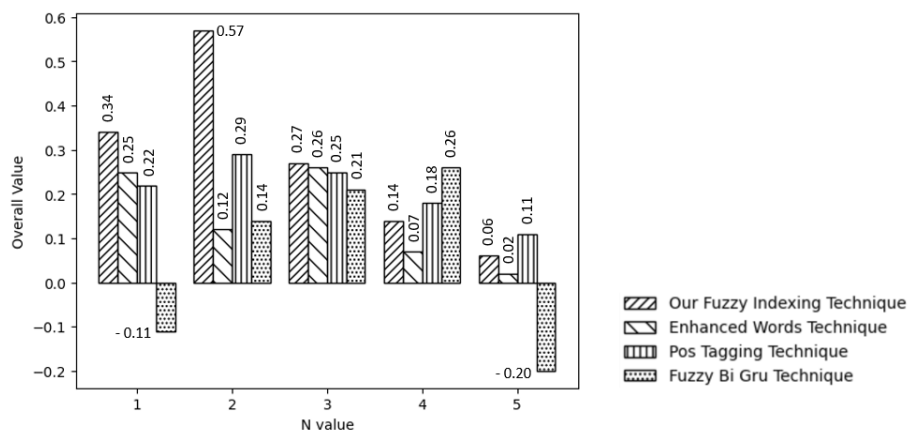


Figure 5. The overall value representation of our fuzzy indexing technique compared to other indexing techniques per N value

4. CONCLUSION AND FUTURE WORK

In conclusion, our work aims to determine the most effective technique for indexing textual documents and to establish a reliable evaluation method for indexes in a textual document corpus using fuzzy logic. We have introduced the concept of fuzzy logic as a key component in the indexing and recognition process for textual data, along with the graph-based N-gram word technique and our novel indexing algorithm. Through practical experiments and case studies using real examples of textual documents, we have examined the impact of different values of n on index selection in the graph-based N-gram approach. We have implemented and discussed the functional aspects of this n -value-dependent indexing technique. Furthermore, we conducted a study to identify the most convincing technique based on precision value, relevance degree, membership degree, and the number of N-grams. The results of our experimental studies led us to infer a fuzzy logic-based function that incorporates multiple criteria, such as the value of N and the degree-value relationship, enabling us to evaluate the set of generated keywords effectively, which helps directly in reducing the imprecision and variability inherent in the traditional comparing methodologies. This gives rise to another problem from a novel perspective: could we manage to apply the same principles for the compound word index case?




REFERENCES

- [1] S. R. El-Beltagy and A. Rafea, "KP-Miner: a keyphrase extraction system for English and Arabic documents," *Information Systems*, vol. 34, no. 1, pp. 132–144, Mar. 2009, doi: 10.1016/j.is.2008.05.002.
- [2] H. Li, J. Zhu, J. Zhang, C. Zong, and X. He, "Keywords-guided abstractive sentence summarization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8196–8203, Apr. 2020, doi: 10.1609/aaai.v34i05.6333.
- [3] D. Buscaldi, G. Felhi, D. Ghoul, J. Le Roux, G. Lejeune, and X. Zhang, *Calcul de similarité entre phrases : quelles mesures et quels descripteurs ? (Sentence Similarity : a study on similarity metrics with words and character strings)*. ATALA et AFPCP, 2020.
- [4] R. K. Mishra, G. Y. S. Reddy, and H. Pathak, "The understanding of deep learning: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–15, Apr. 2021, doi: 10.1155/2021/5548884.
- [5] P. Yang, Y. Ge, Y. Yao, and Y. Yang, "GCN-based document representation for keyphrase generation enhanced by maximizing mutual information," *Knowledge-Based Systems*, vol. 243, May 2022, doi: 10.1016/j.knosys.2022.108488.
- [6] N. Nikzad-Khasmakhi *et al.*, "Phraseformer: multimodal key-phrase extraction using transformer and graph embedding," *Prepr. arXiv.2106.04939*, Jun. 2021, [Online]. Available: <http://arxiv.org/abs/2106.04939>
- [7] E. Papagiannopoulou and G. Tsoumakas, "A review of keyphrase extraction," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 2, Mar. 2020, doi: 10.1002/widm.1339.
- [8] X. Li and M. Daoutis, "Unsupervised key-phrase extraction and clustering for classification scheme in scientific publications," *Prepr. arXiv.2101.09990*, Jan. 2021.
- [9] S. Siddiqi and A. Sharan, "Keyword and keyphrase extraction techniques: a literature review," *International Journal of Computer Applications*, vol. 109, no. 2, pp. 18–23, Jan. 2015, doi: 10.5120/19161-0607.
- [10] L. Chi and L. Hu, "ISKE: an unsupervised automatic keyphrase extraction approach using the iterated sentences based on graph method," *Knowledge-Based Systems*, vol. 223, Jul. 2021, doi: 10.1016/j.knosys.2021.107014.
- [11] F. Boudin, "Unsupervised keyphrase extraction with multipartite graphs," *Prepr. arXiv.1803.08721*, Mar. 2018.
- [12] Y. Sun, H. Qiu, Y. Zheng, Z. Wang, and C. Zhang, "SIFRank: a new baseline for unsupervised keyphrase extraction based on pre-trained language model," *IEEE Access*, vol. 8, pp. 10896–10906, 2020, doi: 10.1109/ACCESS.2020.2965087.
- [13] H. Ye and L. Wang, "Semi-supervised learning for neural keyphrase generation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4142–4153. doi: 10.18653/v1/D18-1447.
- [14] X. Li, P. Lu, L. Hu, X. Wang, and L. Lu, "A novel self-learning semi-supervised deep learning network to detect fake news on social media," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19341–19349, Jun. 2022, doi: 10.1007/s11042-021-11065-x.




- [15] O. Terrada, S. Hamida, B. Cherradi, A. Raihani, and O. Bouattane, "Supervised machine learning based medical diagnosis support system for prediction of patients with heart disease," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, pp. 269–277, 2020, doi: 10.25046/aj050533.
- [16] O. Terrada, B. Cherradi, A. Raihani, and O. Bouattane, "Atherosclerosis disease prediction using supervised machine learning techniques," in *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Apr. 2020, pp. 1–5, doi: 10.1109/IRASET48871.2020.9092082.
- [17] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 670–680, doi: 10.18653/v1/D17-1070.
- [18] M. Xia, A. Anastasopoulos, R. Xu, Y. Yang, and G. Neubig, "Predicting performance for natural language processing tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8625–8646, doi: 10.18653/v1/2020.acl-main.764.
- [19] J. Ye, T. Gui, Y. Luo, Y. Xu, and Q. Zhang, "One2Set: generating diverse keyphrases as a set," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4598–4608, doi: 10.18653/v1/2021.acl-long.354.
- [20] H. Tsukagoshi, R. Sasano, and K. Takeda, "Comparison and combination of sentence embeddings derived from different supervision signals," in *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, 2022, pp. 139–150, doi: 10.18653/v1/2022.starsem-1.12.
- [21] S. Varastehpour, H. Sharifzadeh, and I. Ardekani, "A comprehensive review of deep learning algorithms," *Unitec ePress Occasional and Discussion Papers Series*, New Zealand, 2021, doi: 10.34074/ocds.092.
- [22] H. Abdel-Jaber, D. Devassy, A. Al Salam, L. Hidaytallah, and M. EL-Amir, "A review of deep learning algorithms and their applications in healthcare," *Algorithms*, vol. 15, no. 2, Feb. 2022, doi: 10.3390/a15020071.
- [23] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747v2*, Sep. 2016, [Online]. Available: <http://arxiv.org/abs/1609.04747>
- [24] R. Campos, et al., "Yake! collection-independent automatic keyword extractor," *European Conference on Information Retrieval*, Springer, Cham, 2018, doi: 10.1007/978-3-319-76941-7_80.
- [25] M. Won, B. Martins, and F. Raimundo, "Automatic extraction of relevant keyphrases for the study of issue competition," *Proceedings of the 20th international conference on computational linguistics and intelligent text processing*, 2019, doi: 10.1007/978-3-031-24340-0_48.
- [26] Y. Yu, and V. Ng, "Wikirank: improving keyphrase extraction based on background knowledge," arXiv preprint, 2018, doi: 10.48550/arXiv.1803.09000.
- [27] J. Chen, X. Zhang, Y. Wu, Z. Yan, and Z. Li, "Keyphrase generation with correlation constraints," arXiv preprint, 2018, doi: 10.48550/arXiv.1808.07185.
- [28] K. Ethayarajh, "Unsupervised random walk sentence embeddings: a strong but simple baseline," *Proceedings of The Third Workshop on Representation Learning for NLP*, 2018, doi: 10.18653/v1/W18-3012.
- [29] A. Vaswani, et al. "Attention is all you need," *Advances in neural information processing systems* 30, 2017, doi: 10.48550/arXiv.1706.03762.
- [30] S. Hamida, B. Cherradi, O. Terrada, A. Raihani, H. Ouajji, and S. Laghmati, "A novel feature extraction system for cursive word vocabulary recognition using local features descriptors and gabor filter," in *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, Sep. 2020, pp. 1–7, doi: 10.1109/CommNet49926.2020.9199642.
- [31] L. Ajalloula, K. Najmani, A. Zellou, and E. H. Benlahmar, "Doc2Vec, SBERT, InferSent, and USE which embedding technique for noun phrases?," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Mar. 2022, pp. 1–5, doi: 10.1109/IRASET52964.2022.9738300.
- [32] L. Rassam, C. Aldiebesghanem, A. Zellou and E. B. Lahmar, "Fuzzy logic-based N-gram graph technique for evaluating textual documents indexes," *2022 4th International Conference on Computer Communication and the Internet (ICCCI)*, Chiba, Japan, 2022, pp. 78-82, doi: 10.1109/ICCCI55554.2022.9850268.
- [33] M. Krapivin, A. Autaeu, and M. Marchese, "Large dataset for keyphrases extraction," UNSPECIFIED, 2009. [Online]. Available: <http://eprints.biblio.unitn.it/1671/>.
- [34] Y. Gallina, F. Boudin, and B. Daille, "KPTimes: a large-scale dataset for keyphrase generation on news documents," in *Proceedings of the 12th International Conference on Natural Language Generation*, 2019, pp. 130–135, doi: 10.18653/v1/W19-8617.
- [35] T. D. Nguyen, and M. Y. Kan, "Keyphrase extraction in scientific publications," *International conference on Asian digital libraries*, Springer, Berlin, Heidelberg, 2007, doi: 10.1007/978-3-540-77094-7_41.
- [36] S. N. Kim, O. Medelyan, M. Y. Kan, T. Baldwin, and L. P. Pingar, "SemEval-2010 task 5: automatic keyphrase extraction from scientific," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, 2010.
- [37] F. Liu, X. Huang, W. Huang, and S. X. Duan, "Performance evaluation of keyword extraction methods and visualization for student online comments," *Symmetry*, vol. 12, no. 11, 2020, doi: 10.3390/sym12111923.

BIOGRAPHIES OF AUTHORS






Latifa Rassam    She received her Eng. degree in Web Engineering and Mobile Computing from the High National School of Computer Science and System Analysis (ENSIAS) at Mohammed V University in Rabat, Morocco in 2018. She is a Ph.D. student in the software project management (SPM) Research Team at ENSIAS, and at Mohammed V University in Rabat. Her research interests include primarily natural language processing, data indexing, and the internet of things (IoT) domains where she is the author/coauthor of over 5 research publications. She can be contacted at email: rassamlatifa@gmail.com.






Imane Ettahiri    She received her Eng. degree in Software Engineering from the High National School of Computer Science and System Analysis (ENSIAS) at Mohammed V University in Rabat, Morocco in 2014. Currently, she is a Ph.D. student in the Alqualsadi Research Team at ENSIAS, and at Mohammed V University in Rabat. Additionally, she is a part-time professor at Mohammed V University. Her research interests include dynamic enterprise architecture, its quality, integration, and evaluation. She can be contacted at email: i.ettahiri@gmail.com.



Ahmed Zellou    received his Ph.D. in Applied Sciences at the Mohammedia School of Engineers, Mohammed V University, Rabat, Morocco, in 2008, and his habilitation to supervise research work in 2014. He became a full professor in 2020. His research interests include interoperability, mediation systems, distributed computing, data, indexing, recommender systems, data quality, and semantic web where he is the author/coauthor of over 100 research publications. He can be contacted at email: ahmed.zellou@um5.ac.ma.



Karim Doumi    he is an Associate Professor at Mohammed V University in Rabat, Morocco. He received his Ph.D. in Software Engineering from the Alqualsadi Research Team at the High National School of Computer Science and System Analysis (ENSIAS), Mohammed V University in Rabat, in 2013. His research interests include enterprise architecture, adaptability, agility in enterprise architecture, and business IT alignment. He can be contacted at email: k.doumi@um5r.ac.ma.