STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, April 2026, pp 2796–2818.
Published online in International Academic Press (www.IAPress.org)

Applied the Fuzzy Naïve Bayes Algorithm to Word Sense Disambiguating

Bouchra DAOUDI*, Hassania HAMZAOUI

L2MASI, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah university, FEZ, MOROCCO

Abstract In this article, the Fuzzy Naïve Bayes algorithm is presented. This algorithm integrates the classical Naïve Bayes model with fuzzy logic, in order to address the complex problem of semantic disambiguation in Arabic. This task remains particularly challenging due to the morphological richness and lexical ambiguity of the Arabic language. The approach adopted aims to model the uncertainty linked to the multiple possible interpretations of a word in context by assigning each meaning a fuzzy degree of membership rather than a strict classification. The evaluation of the algorithm was conducted on three distinct corpora, utilising lexical and syntactic features. The performance obtained was systematically compared with that of the standard Naïve Bayes model. The experimental results demonstrate a substantial enhancement in terms of accuracy and robustness, underscoring the contribution of fuzzy logic to the management of semantic uncertainties.

Keywords Fuzzy logic, Naïve Bayes Algorithm, Word Sense Disambiguation, Arabic language.

DOI: 10.19139/soic-2310-5070-3089

1. Introduction

Natural language processing (NLP) [34] is a core area of artificial intelligence. The aim of this field is to enable machines to understand, generate, and interact fluidly in human language. The objective of this study is to establish a connection between human communication and computational interpretation by emulating human linguistic capabilities in various domains, including text comprehension, machine translation and human-machine interaction [36].

A fundamental challenge in natural language processing is word sense disambiguation (WSD) [37], defined as the task of automatically identifying the correct meaning of a word with multiple meanings based on its context of use. While this process is generally intuitive for humans, it remains extremely complex for automatic systems. WSD has been shown to play a crucial role in a variety of applications, including machine translation, information retrieval, text understanding and large generative models such as large language models [38].

The challenge of this undertaking is further compounded when applied to morphologically complex languages such as Arabic [29], which is characterised by nonconcatenative morphology, an orthography devoid of explicit short vowels, and extensive dialectal variation [10, 12]. The Ambiguity in Arabic can occur at several levels, including the syntactic, semantic and pronominal/anaphoric levels. Despite considerable advances in WSD approaches for languages such as English, research in this field for Arabic remains limited, primarily due to the paucity of annotated linguistic resources and accessible corpora. Furthermore, despite substantial advancements in machine learning and artificial intelligence, including deep learning, short-term learning, explainable AI, etc [39, 58], these techniques necessitate substantial

*Correspondence to: Bouchra DAOUDI (Email: bouchra.daoudi@usmba.ac.ma). L2MASI, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah university, FEZ, MOROCCO

ISSN 2310-5070 (online) ISSN 2311-004X (print)

amounts of annotated data, which is not readily available for Arabic. This hinders their large scale deployment.

From English to many other languages, such as Arabic, a variety of approaches to solving WSD problems have been proposed [40]. These methodologies are predicated on rigorous mathematical models that aspire to furnish precise decisions. However, human language is inherently vague and subjective, with meanings contingent on context, individual interpretation, and cultural context. The inherent tension between the pursuit of algorithmic precision and the presence of linguistic ambiguity has been identified as a significant factor that impedes the efficacy of purely logical or statistical models [41]. This is the background to the interest in fuzzy logic [16]. Proposed by Zadeh in 1965 [17], fuzzy logic makes it possible to represent uncertainty, imprecision, and gradual reasoning, which are typical of human reasoning. In the context of WSD, it offers an alternative framework for modelling degrees of membership of different senses, rather than forcing a binary or exclusive classification. In recent years, a number of studies have demonstrated the usefulness of this approach for overcoming the limitations of traditional methods and better representing the uncertain nature of natural language.

The following proposal is put forward: a lexical disambiguation method based on the Fuzzy Naïve Bayes algorithm (FNB). This model synthesises the simplicity and probabilistic efficiency of the Naïve Bayes classifier [19] with the flexibility of fuzzy logic, with a view to more effectively managing the linguistic complexity of Arabic. This integration has been demonstrated to enhance the system's capacity to discern the accurate meaning of a word in ambiguous contexts, thereby accounting for the vague and nuanced nature of natural language.

The structure of the paper is as follows. Section 2 provides a brief review of the literature on the use of the Naïve Bayes classifier for WSD. Section 3 presents a background on the Arabic language and fuzzy set theory, including fuzzy event probabilities and fuzzy clustering. This is followed in Section 4 by the theoretical foundations of the main approach, covering both the Naïve Bayes classifier and its fuzzy version. Section 5 provides details of the methodology, including the data pre-processing and feature extraction steps, followed by the use of word embeddings and fuzzy C-Means clustering (FCM) [20]. The following sections are then devoted to an explanation of the training and disambiguation phases. The experiments and the results are then discussed in Section 6, including a description of the dataset, the experimental setup and the approach limitations. In conclusion, the final section of the study summarises the results and puts forward a series of recommendations for future research.

2. Word Sense Disambiguation: From Naïve Bayes to Fuzzy Logic

This section provides an overview of previous studies relevant to word sense disambiguation, with a focus on two primary approaches. The first is a multilingual WSD algorithm based on the Naïve Bayes model. The second is the application of fuzzy logic techniques in natural language processing for WSD.

2.1. The multilingual WSD algorithm by Naïve Bayes algorithm

The Naïve Bayes classifier [19] has been used in much of the research on WSD. This simple yet powerful probabilistic approach has proven its effectiveness, especially when enriched with relevant linguistic features. The following is a summary of some representative contributions, grouped by language studied.

The English language:

Shimazu & Le (2005) [3] have shown that the integration of various linguistic features such as collocations, syntactic dependencies, etc., enables the Naïve Bayes classifier to achieve high accuracy. Their pioneering study highlighted the importance of enriching linguistic representations to improve lexical disambiguation performance.

More recently, Abraham et al. (2024) [1] developed a Naïve Bayes-based system to disambiguate ambiguous words in English. Their model relies on collocational feature extraction to predict the grammatical category of words (noun, verb, adjective, adverb). The system achieves satisfactory performance with an F1-measure of 78%.

The Arabic language:

Elmougy et al. (2008) [4] proposed an Arabic word disambiguation method based on the rootization algorithm and the Naïve Bayes classifier. Their approach focuses on resolving non-diacritical word ambiguities, which is a common problem in Arabic due to the absence of diacritics in many digital documents.

Eid et al. (2010) [6], although focused on the Rocchio classifier, provide a relevant comparative analysis with Naïve Bayes, using an Arabic lexical dataset. Their study highlights the strengths and limitations of supervised approaches.

Merhben et al. (2012) [7] conducted an experimental evaluation of several WSD techniques for Arabic, including Naïve Bayes algorithm, decision list, and k-nearest neighbour. They emphasize the difficulties induced by the morphological richness of the language.

Hadni et al. (2016) [8] proposed a word disambiguation method to improve the categorisation of Arabic texts. Their approach uses Naïve Bayes and relies on the use of external resources, such as Arabic WordNet and WordNet, to combine semantic relations in the same local context. The results show that the application of this method significantly improves the performance of the categorisation system.

The Hindi language:

Singh et al. (2014) [9] used the Naïve Bayes classifier for word disambiguation (WSD) in Hindi using eleven features, including local context, collocations, unordered word lists, nouns and vibhaktis. Experimental results showed that the addition of enriched features significantly improved the accuracy of the model, reaching 86.11% for nouns after applying morphology, compared to 77.52% for unordered word lists. These studies demonstrate that Naive Bayes remains a robust and flexible baseline for WSD. It offers a straightforward yet effective probabilistic framework that can handle multilingual data and deliver competitive performance, despite its independence assumptions.

2.2. Fuzzy Logic in NLP and its Use in WSD

Beyond probabilistic models, a substantial body of research has explored the use of fuzzy logic to address the ambiguity and gradience inherent in natural language. Early foundational work showed that fuzzy set theory offers a systematic approach to modelling linguistic semantics, facilitating the interpretation of vague lexical units such as nouns, adjectives, adverbs and conjunctions [62]. More recent surveys confirm the increasing use of fuzzy theories in NLP tasks, emphasising their capacity to quantify conceptual vagueness in issues related to cognition, translation and semantic comprehension [64, 63]. Within the specific domain of WSD, fuzzy logic has been applied through multiple paradigms, including fuzzy inference systems, fuzzy graphs, fuzzy semantic similarity, and fuzzy clustering techniques. Several studies have shown that fuzzy modelling can enhance sense induction and disambiguation by representing senses as linguistic variables and assigning graded memberships to possible interpretations [65, 66]. Fuzzy graphs have been used to weight semantic relations in WordNet according to their importance, thereby improving the identification of intended meanings [67]. Extended fuzzy WordNets have also demonstrated superior performance in Hindi WSD through fuzzy relation composition and graph connectivity measures [68]. Other studies use fuzzifiers, fuzzy classifiers and fuzzy similarity models to address lexical ambiguity in morphologically rich languages such as Arabic [69] and Hindi, employing high-dimensional contextual representations combined with Fuzzy C-Means clustering [70]. Further work demonstrates that fuzzy clustering of semantic features can effectively reveal latent meanings in complex web documents and facilitate soft semantic categorisation, thereby reinforcing the importance of fuzzy approaches for sense discrimination [71]. Collectively, these contributions confirm that fuzzy logic is an effective way of modelling uncertainty and capturing nuanced semantic relations. This makes it a powerful alternative to, or complement for, classical statistical and machine learning methods in WSD.

3. Background: Arabic language and Fuzzy set theory

In this section, we review the foundational concepts and definitions necessary to understand the proposed approach.

We provide an overview of the key characteristics of the Arabic language [10], emphasizing its morphological richness and the challenges it poses for natural language processing tasks [25]. Furthermore, we introduce the principles of fuzzy set theory [14], which serve as the basis for handling the inherent uncertainty and ambiguity in word sense disambiguation. These preliminary insights, definitions, and relevant results establish the theoretical framework for this article.

3.1. Arabic language

Arabic is one of the oldest Semitic languages in the world. This language is both difficult and interesting. It is interesting because of its history, the strategic importance of its people and the region they occupy, and its cultural and literary heritage. It also represents a challenge because of its complex linguistic structure [11].

The Arabic language has three main varieties:

1. Classical Arabic language (CLA) is used in religious texts and in many ancient Arabic manuscripts.
2. Modern Standard Arabic (MSA) is the language of formal communication understood by the majority of Arabic speakers and is commonly used on the radio, in newspapers, and on television.
3. Dialectal or colloquial Arabic is used in everyday conversation and, more recently, on television and radio.

In Arabic, in MSA, and also in CLA, there are two main types of sentence: a verbal sentence (the sentence begins with a verb) and a noun sentence (the sentence begins with a noun) [12].

Sentence structure analysis in Arabic is considered the most complex because of the flexibility of word order, morphological complexity, the fact that Arabic is a clitic or clitic-directed language (Arabic words are derived), the omission of diacritics, the frequent production of homographs of words with or without the same pronunciation, and the fact that Arabic is a pro-drop language. In short, this complexity is due to the following aspects, which are morphological, spelling, dialects, short vowels and word order. For these reasons, Arabic is an ambiguous language. There are two main levels of Arabic ambiguity: homograph and polysemy.

- Homographs are words that have the same spelling but different meanings.

- Polysemy is the association of one word with more than one meaning.

Ambiguity in Arabic can be also present in other levels, such as internal word structure ambiguity, syntactic ambiguity, semantic ambiguity, constituent boundary ambiguity, and anaphoric ambiguity [13].

Research in Arabic WSD is very limited due to the lack of resources available, such as corpora, dictionaries, and datasets suitable for computing tasks [29].

3.2. Fuzzy Set Theory

The foundations of fuzzy logic have evolved significantly since its inception, leading to a substantial expansion in its applications and a growing influence on the basic sciences. From artificial intelligence to control systems, fuzzy logic has proven to be an indispensable tool for modeling complex systems and decision-making processes [15].

Fuzzy logic was introduced in the 1960s by Dr. Lotfi Zadeh [17] as an extension of classical logic. It is fundamentally based on the concept of fuzzy sets, which generalises the notion of classical (crisp) sets. Unlike traditional binary logic, which operates on absolute true or false values (1 or 0), fuzzy logic is grounded in the idea of ”degrees of truth.” Objects in fuzzy logic can belong to multiple subsets with varying degrees of membership, represented as values in the interval $[0, 1]$. This allows fuzzy logic to model uncertainty and vagueness in a way that closely mirrors human perception and reasoning.

3.2.1. Fuzzy Set: Zadeh defines a fuzzy set as one in which membership is determined by a graded characteristic function. This function assigns each object a degree of membership, reflecting the degree to which the object belongs to the set. This approach enables nuanced representation of uncertainty, where membership is not binary but continuous [14].

Formally, a fuzzy subset F of a reference set X is characterized by a membership function σ_F that associates with each element $x \in X$ a membership degree $\sigma_F(x) \in [0, 1]$. This is expressed as:

$$\forall x \in X, \sigma_F(x) \in [0, 1]$$

Here, the value of $\sigma_F(x)$ represents the degree to which the element x belongs to the fuzzy subset F. For instance, in a fuzzy set representing "tall people," the membership function might assign higher values to taller individuals and lower values to shorter ones, reflecting a gradual transition rather than a sharp boundary.

The flexibility of fuzzy sets and their ability to handle imprecise data make them an invaluable tool in various domains, including decision-making systems, control theory, and natural language processing.

3.2.2. Type of membership function: The membership functions are curves that represent the degree to which an element belongs to a fuzzy set. The most common shapes include triangular and trapezoidal functions, which are simple, easy to interpret, and ideal when the boundaries between categories are approximately linear. Gaussian and bell-shaped functions offer smoother transitions while remaining suitable when the boundaries can be approximated linearly around the core of the distribution. Conversely, sigmoid functions are well-suited to gradual or asymmetrical transitions, especially when the boundaries between categories, though not rigid, exhibit a predominantly linear trend[59].
These functions are of fundamental importance, insofar as they determine how data is represented, weighted and interpreted in a fuzzy system. A satisfactory definition of these curves facilitates the accurate capture of uncertainty, the modelling of the fuzzy boundaries between classes and the substantial enhancement of the quality of classification and fuzzy reasoning.

3.3. Probability of fuzzy event

The concept of the probability of a fuzzy event was introduced by Zadeh in [42], extending the classical probability framework to incorporate fuzzy sets. This approach allows the probabilistic evaluation of events with imprecise boundaries, combining the ideas of fuzzy logic and traditional probability.
Formally, let B be a σ-field of Borel subsets in $\mathbb{R}^n$ and P be a probability measure over the space Ω. Let Z be a fuzzy event in B. Thus, the probability of Z can be expressed as a Lebesgue-Sieltjes integral [43]:

$$P(Z) = \int_{Z \subseteq \mathbb{R}^n} dP = \int_{Z \subseteq \mathbb{R}^n} \mu_Z(x) dP = E\left(\mu_Z\right) \tag{1}$$

where $\mu_Z(x)$ represents the membership function of the fuzzy set Z.

Thus, the probability of a fuzzy event Z is defined as the mathematical expectation of its membership function. This can also be expressed in terms of the probability density function $P(x)$ as:

$$P(Z) = \int_{Z \subseteq \mathbb{R}^n} \mu_Z(x) P(x) dx \tag{2}$$

This formulation bridges the gap between fuzzy logic and probability theory, enabling the modeling of events with uncertainty both in their definitions and outcomes.

3.4. Fuzzy clustering

Clustering is the process of grouping the data into classes or clusters. Clustering methods can be broadly divided into two categories: hierarchical methods (creates a hierarchy of clusters by merging or splitting them based on similarity measures) and partitioning methods (divides data into several disjoin subsets), and the latter can itself be divided into two groups: hard clustering and soft clustering. The hard clustering is where each element belongs to only one cluster, while the soft clustering is where each element belongs in more than one cluster.

Fuzzy clustering [18] is a soft clustering method that has gained popularity for data analysis. In fuzzy clustering, each data point can belong to multiple clusters, with a set of membership coefficients indicating the degree of association with each cluster. Among the most widely used methods is fuzzy c-means, which generalizes the well-known k-means algorithm.

4. Theoretical Foundation of the Main Approach

This section provides the theoretical underpinnings of the approach utilised in this study. It outlines the key principles that form the basis for our methodology. By examining the theoretical framework in depth, we aim to establish a clear understanding of the core concepts and their relevance to solving the problem at hand. The goal is to ensure that the reader grasps the rationale behind the chosen approach and its theoretical advantages, which guide the subsequent implementation and experiments.

4.1. Naïve bayes algorithm

The Naïve Bayes (NB) algorithm is one of the most commonly employed supervised classification algorithms, utilized across diverse fields, particularly in NLP, where it is extensively applied for text classification tasks involving high-dimensional training datasets [19].

The NB classifier is a straightforward probabilistic-based classifier, founded on Bayes' theorem.

Formally, let the sample space $\Omega = \{1, \ldots, C\}$ where C is the total number of classes. Let $X = \{X_1, X_2, \ldots, X_n\}$ a random vector of data and $w_i, i \in \Omega$ is the class in space of decision for the vector X. So, the probability of the class w_i, given the vector X, can be estimated using the Bayes Theorem:

$$P\left(w_i \mid X\right) = \frac{P\left(X \mid w_i\right) P\left(w_i\right)}{P(X)} \tag{3}$$

In informal terms, this can be written as follows:

$$posterior = \frac{likelihood \times prior}{evidence}$$

Such that: – $P\left(w_i \mid X\right)$: the posterior probability, i.e., the probability that the class is w_i given the observed data X.

- $P\left(X \mid w_i\right)$: the likelihood , i.e., the probability of observing the data X assuming that the class is w_i.
- $P\left(w_i\right)$: the prior, i.e., the initial probability of class w_i, independent of the data.
- $P(X)$: the evidence, i.e., the total probability of observing the data X, across all possible classes.

The NB algorithm assumes that all X_i, $i = 1, \ldots, n$, are independent of each other, so we have:

$$\begin{aligned} P\left(w_i \mid X\right) &= \frac{P\left(X_1, X_2, \ldots, X_n \mid w_i\right) P\left(w_i\right)}{P(X)} \\ &= \frac{1}{S} P\left(w_i\right) \prod_{k=1}^{n} P\left(X_k \mid w_i\right) \end{aligned} \tag{4}$$

Such that: $S = \sum\limits_{i=1}^{C} P\left(w_i\right) \prod\limits_{k=1}^{n} P\left(X_k \mid w_i\right)$

Since S acts as a normalization constant, it ensures that probabilities sum to one. However, it is not required for the classification decision since it cancels out when comparing probabilities across classes. Therefore, S is a constant (scale factor), and we can dispense with it. The probability $P\left(w_i \mid X\right)$ is therefore given by :

$$P\left(w_i \mid X\right) = P\left(w_i\right) \prod_{k=1}^{n} P\left(X_k \mid w_i\right) \tag{5}$$

The assessment rule for the Naïve Bayes Network is given by:

”Select class w_i for the vector X if $P\left(w_i \mid X\right) \geq P\left(w_j \mid X\right) \quad$ for all $i \neq j \quad i, j \in \Omega$”

In the context of Arabic WSD, the NB algorithm is used to predict the sense of an ambiguous word based on features extracted from the context. In other words, it is used to calculate the conditional probabilities of the different meanings of an ambiguous word, based on features extracted from its context. It then ranks the senses according to the highest probability, based on the assumption of feature independence, which simplifies its implementation and calculations [1].

The NB algorithm is presented as follows: let v the ambiguous word, $SNS = \{sns_1, ..., sns_J\}$ be the set of word senses and $\Phi = \{\phi_1, ..., \phi_K\}$ be the set of features.

We look to determine the appropriate sense of the ambiguous word in the context, i.e., the correct sense is sns_I, such that $P\left(sns_I \mid \Phi\right) > P(sns_J \mid \Phi)$ for $sns_I \neq sns_J$, and $sns_I, sns_K \in SNS$.

We will use the logarithm to make the computation simpler. Then we try to assign v to the sense sns_I where:

$$sns_I = \underset{sns_J \in SNS}{\arg\max} \left(\log\left(P\left(\Phi \mid sns_J\right)\right) + \log\left(P\left(sns_J\right)\right)\right) \tag{6}$$

With the NB assumption: $P\left(\Phi \mid sns_J\right) = \prod_{k=1}^{K} P\left(\phi_k \mid sns_J\right)$, so sns_I can be rewritten as:

$$sns_I = \underset{sns_J \in SNS}{\arg\max} \left(\sum_{k=1}^{K} \log\left(P\left(\phi_k \mid sns_J\right)\right) + \log\left(P\left(sns_J\right)\right)\right) \tag{7}$$

By applying this approach, we can effectively predict the most likely sense of the ambiguous word based on its contextual features, making the NB algorithm a powerful tool for WSD in NLP.

4.2. Fuzzy Naïve Bayes

The Fuzzy Naïve Bayes algorithm [44] is grounded in Zadeh’s definition of the probability of fuzzy events [42]. By applying this principle, a formal methodology for FNB can be derived, particularly useful for AWSD in our case, where the goal is to identify the correct sense of an ambiguous word in a given context.

Formally, from the equation 2 and 4 and under the assumption that each feature ϕ_k is conditionally independent of every other feature ϕ_l, for all $k \neq l \leq K$, the Fuzzy Naïve Bayes algorithm can be expressed as:

$$P\left(sns_i \mid \phi_1, \phi_2, \ldots, \phi_K\right) = \frac{P\left(sns_i\right) \cdot \prod_{k=1}^{K}\left(\mu_{sns_i}\left(\phi_k\right) \cdot P\left(\phi_k \mid sns_i\right)\right)}{S}$$

Once again, to simplify calculations and avoid numerical issues associated with handling very small probability products, we will use the logarithm. This allows us to transform products into sums, making computations more stable and faster to execute. Thus, the original expression of the FNB becomes:

$$\begin{aligned} \log\left(P\left(sns_i \mid \phi_l, \phi_2, \ldots, \phi_K\right)\right) &= \log(1/S) + \log P\left(sns_i\right) \\ &+ \sum_{k=1}^{K}\left(\log\left(\mu_{sns_i}\left(\phi_k\right)\right) + \log\left(P\left(\phi_k \backslash sns_i\right)\right)\right) \end{aligned} \tag{8}$$

As in the NB method, the parameters for $P\left(\phi_k \backslash sns_i\right)$ and $\mu_{sns_i}(\bullet)$ are learned from the data. Again, S is a scale factor, and it is not necessary to be computed in the classification rule for FNB.
The FNB classification for AWSD can be expressed as:

$$sns_I = \underset{sns_J \in SNS}{\arg\max} \left(\sum_{k=1}^{K} \left(\log P\left(sns_i\right) + \log\left(\mu_{sns_i}\left(\phi_k\right)\right) + \log\left(P\left(\phi_k \backslash sns_i\right)\right)\right) \right) \tag{9}$$

such that :

- **Membership Function** $(\mu_{sns_i}(\phi_k))$: Determines the degree of association between the ambiguous word v (in a given sense sns_i) and the contextual feature ϕ_k.
- **Probability Terms**:
  – $P(sns_i)$: Prior probability of the sense sns_i, which could be derived from a corpus or linguistic knowledge.
  – $P(\phi_k \backslash sns_i)$: Likelihood of the contextual feature ϕ_k occurring given the sense sns_i

In practice, to calculate the membership value, we will refer to FCM. The membership values derived from FCM serve as a critical component in our hybrid methodology. These values capture the degree of association between a given context and potential word senses, reflecting the inherent fuzziness in natural language. By integrating this information into the probabilistic framework of Naïve Bayes, we enhance its ability to handle ambiguity and uncertainty effectively.

Our methodology for Arabic word sense disambiguation leverages a hybrid approach, fusing Naïve Bayes and Fuzzy C-Means classification techniques. With the inherent ambiguity present in language, this combination allows us to navigate the complexities of determining the correct sense of a word within its context. Naïve Bayes, known for its simplicity and effectiveness in probabilistic classification, provides a solid foundation for initial classification. Fuzzy C-Means, on the other hand, accommodates the nuanced and often overlapping boundaries between word senses, offering a more flexible framework for clustering and classification. By synergizing these methods, we aim to achieve a more robust and accurate disambiguation process. Our evaluation will compare the performance of this hybrid approach against that of Naïve Bayes alone, aiming to identify the most effective strategy for resolving word sense ambiguity. Ultimately, our goal is to contribute to the advancement of natural language understanding and semantic analysis in this language.

5. Proposed approach

In this section, we present the description of our method for WSD of target words and explain how the most appropriate sense for a target word in a given sentence is determined using FNB algorithm. The proposed method combines the probabilistic framework of Naïve Bayes with fuzzy logic to handle the inherent ambiguity and overlapping boundaries between word senses.

The process begins with a series of preprocessing steps [22]. These steps prepare the input data and convert them into numerical representations suitable for further analysis. Contextual features [2], such as surrounding words and POS tags [24], are extracted to provide meaningful context for the disambiguation task. The membership values, representing the degree of association between each context feature and the potential senses of the target word, are calculated using the FCM clustering algorithm [21]. These values allow for a soft clustering approach, enabling a word to belong to multiple sense clusters with varying degrees of membership. The FNB algorithm then uses these membership values, along with prior probabilities and likelihoods derived from the data, to apply a probabilistic classification rule. This hybrid approach ensures a robust determination of the most likely sense for the target word.

The main framework of our method, as shown in Figure 1, integrates these steps into a cohesive pipeline, demonstrating the synergy between fuzzy logic and probabilistic models in addressing word sense ambiguity.

5.1. Data preprocessing:

Preprocessing is a crucial and foundational task in NLP as it directly impacts the quality and performance of downstream tasks [22]. Text data often contains various special formats such as numbers, dates, punctuation marks, and other non-textual elements. These formats can introduce noise or variability that may affect the effectiveness of the NLP models. Preprocessing involves cleaning and standardizing the data, which may include tasks such as tokenization, lowercasing, removing or normalizing special characters, handling stop words, and converting numerical or date formats into a more interpretable structure. This step ensures that the input data is consistent and suitable for further processing, enabling the model to focus on the relevant linguistic patterns and improve overall accuracy [23].

The preprocessing step in our case comprises the following: Normalization, Removing Stop Words, Tokenization, and Stemming. In Arabic, stemming is particularly challenging due to the language's rich and complex morphology, characterized by extensive inflectional and derivational variations [30]. To address this complexity, several stemming algorithms are employed, which can be broadly categorized into four types: Light Stemmer [45], Rule-Based Stemmer [45], Statistical Stemmer [46], and Artificial Intelligence Approaches [47].

This robust preprocessing framework ensures that the input data are clean, consistent, and suitable for accurate WSD in Arabic, addressing the challenges posed by the language's inherent complexity [35].

5.2. Feature extraction and selection:

The main aim of feature extraction is to transform text of any structure into a list of meaningful keywords or features that can be effectively utilized in supervised learning tasks. Feature extraction is a critical step in text analysis, as it enables the identification of relevant patterns and relationships within the data.

For the Arabic language, various features are commonly considered due to its rich morphological structure and complex syntax. These include morphological features, lexical statistical features, semantic features, and syntactic features. Each of these captures a unique aspect of the language, contributing to a more comprehensive representation of the text.

This paper focuses on two specific approaches to feature extraction: surrounding Words, i.e. the local collocation and part-of-speech tags [24].

By focusing on these features, the proposed approach aims to leverage both lexical and syntactic information to enhance the accuracy and robustness of AWSD.

5.3. Word embedding

Word embedding [26] represents a significant advancement in the field of automatic natural language processing. The methodology under discussion involves the representation of each word by a vector of real numbers in a multidimensional space, with the aim of capturing its syntactic and semantic relationships with other words. This process involves the transformation of raw text into a digital format suitable for utilisation by learning models. Word2Vec is considered to be one of the most well-known methods, and has been established as a benchmark since 2013 [48], despite the fact that the concept was introduced as early as 2003 [49]. Word2Vec [50] is predicated on self-supervised learning, a process which automatically generates meaningful vector representations from unlabelled text. The employment of this representation facilitates enhanced comprehension of the contextual affinities among words, thereby augmenting the efficacy of numerous NLP operations. Word2Vec employs two complementary architectures that learn contextual relationships between words from their co-occurrence in a large corpus [50].

5.4. Fuzzy C-Means Clustering

The primary objective of employing the FCM clustering [21] algorithm in this context is to compute the membership degrees of each word within a multidimensional feature space. These words have already undergone a transformation into numerical representations through the word embedding process, where each word is encoded as a high-dimensional vector capturing its semantic and syntactic properties. FCM is particularly good at carrying out this role, as it matches the fuzzy linguistic character in that a word might belong partially to more than one semantic set or category rather than being exclusively dedicated to a single one. FCM's operation is most useful for activities like Arabic WSD, in that uncertainty and situational subtleties always lead to sets of overlapping word senses.

By calculating membership values, FCM maps every word to a set of numbers that express the degree to which it belongs to different clusters, thus giving its context and meaning a probabilistic interpretation. Such degrees can serve as input for further processes, with linguistic relations being more suitably modeled and disambiguations being achieved more successfully.
The membership value can be achieved through the minimization of the following function:

$$J_p(\Sigma, \Delta) = \sum_{t=1}^{n} \sum_{k=1}^{c} (\sigma_{tk})^p \left\| v_t - \delta_k \right\|^2 \tag{10}$$

With – $V = (v_1, v_2, \ldots, v_n)$ represents the dataset, with v_i denoting the i-th data point.
- $\Delta = (\delta_1, \delta_2, \ldots, \delta_c)$ are the centroids of the clusters, with δ_k is the k^{th} cluster center.
- $\Sigma = (\sigma_{tk})_{n \times c}$ is a fuzzy partition matrix, with $\sigma_{tk} \in [0, 1]$ is the membership degree of data point v_t to the fuzzy cluster k.
- $\|v_t - \delta_k\|$ is the Euclidean norm between v_t and δ_k [61].
- $p > 1$ is the fuzziness parameter. p is used to control the fuzzy degree of membership of each data. A higher p increases the fuzziness, while a lower p reduces it.

To minimize the objective function in Equation 10, we need to iteratively calculate the optimal values of Δ and Σ.
Before starting the optimization, the following parameters need to be set:
- **p** : The fuzziness parameter, which determines the level of cluster overlap;
- ε : the tolerance value, which sets the convergence threshold for stopping the algorithm;
- max_*Iter* : The maximum number of iterations allowed during optimization;
- **c** : the number of cluster to be generated.

After initialising the required parameters, we proceed to the main steps of the algorithm:

- **Step 1.** Initialise σ_{tk} membership degrees by random values in $[0, 1]$ so that it verifies the following condition:

$$\sum_{k=1}^{c} \sigma_{tk} = 1, \forall t = 1, \ldots, n. \tag{11}$$

- **Step 2.** Calculate the centres of the cluster using the following expression :

$$\delta_k = \frac{\sum_{t=1}^{n} \sigma_{tk}^p \cdot v_t}{\sum_{t=1}^{n} \sigma_{tk}^p}, k = 1, 2, \ldots, c. \tag{12}$$

- **Step 3.** Update Σ the membership matrix so that it satisfies the constraint (11) by the expression:

$$\sigma_{tk} = \frac{1}{\sum\limits_{z=1}^{c} \left(\frac{\|v_t - \delta_k\|}{\|v_t - \delta_z\|} \right)^{\frac{2}{p-1}}}, t = 1, \ldots, n \text{ and } k = 1, \ldots, c. \tag{13}$$

Steps 2 and 3 are repeated until the number of iterations reaches max_*Iter* or the difference between the current membership matrix and the previous membership matrix becomes less than the tolerance

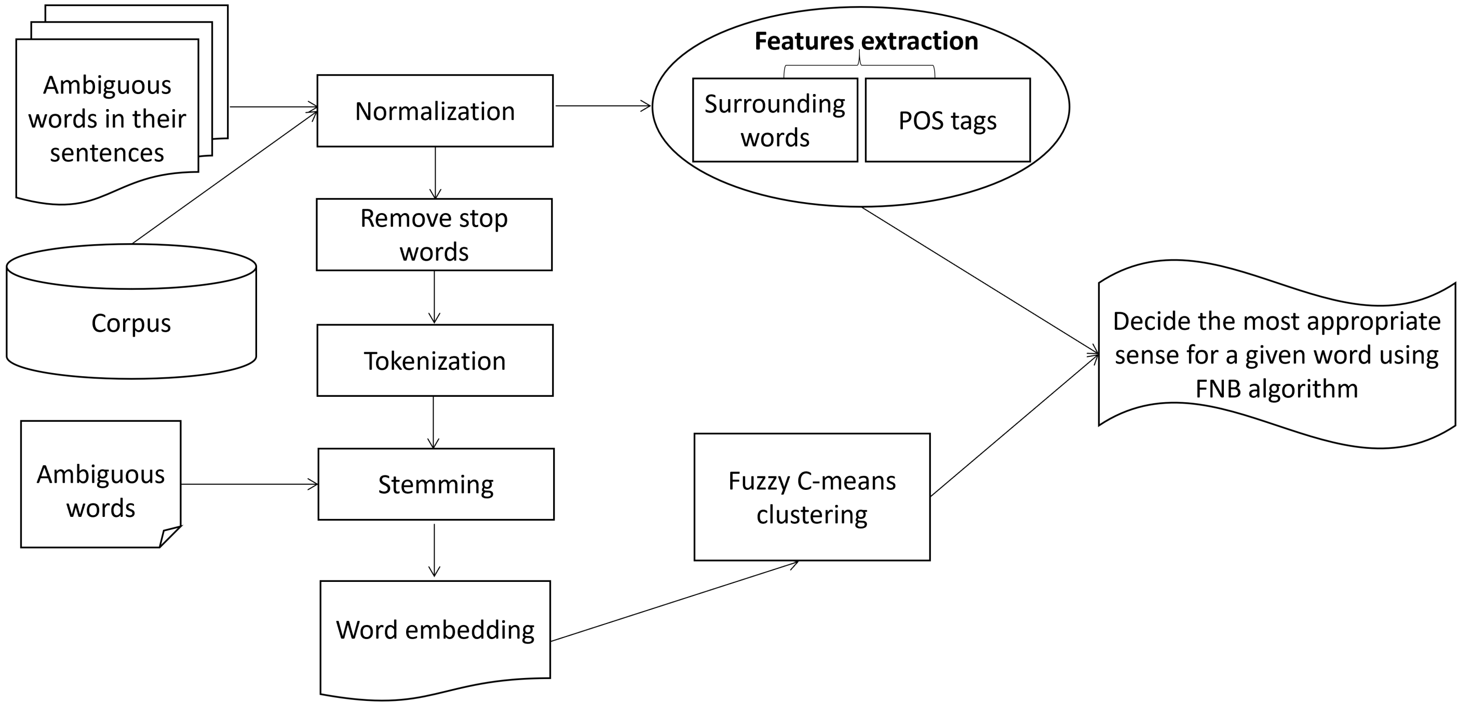


Figure 1. Flowchart of our method

value ε.

However, subsequent to the optimisation stage, the algorithm progresses to a process of cluster validation. The objective of this stage is twofold: firstly, to ascertain the optimal number of clusters, c, and secondly, to determine the fuzzification coefficient, m. In addition, this stage enables an assessment of the compactness and separation of the clusters. The Xie and Beni index (XB) [60] is a particularly widespread index that is utilised in a variety of contexts. It is a multifaceted index that incorporates both membership degrees and data structure to calculate intra-cluster compactness and inter-cluster separation. A superior partition is characterised by low compactness and high separation, and the optimal value of the index corresponding to its minimum facilitates the identification of the optimal number of clusters and fuzzification coefficient.

5.5. Training and Disambiguation Phases

The two primary phases of the AWSD process are delineated in this section: the training phase and the disambiguation phase. During the training phase, the system employs an annotated corpus to learn the relationships between linguistic contexts and the possible meanings of an ambiguous word. In particular, it calculates the membership degrees, the prior probabilities of senses and the probability of contextual feature. Utilising this accumulated knowledge, the system subsequently engages the disambiguation phase algorithm during its analysis of a new, unlabelled context. This process entails the combination of the elements learned to identify the most likely meaning of the target word, thus facilitating effective automation.

5.5.1. The Training Phase: The training phase constitutes the preparatory stage, during which the system acquires the capacity to associate linguistic contexts with the appropriate senses of an ambiguous word. The algorithm extracts contextual features (surrounding words and POS tags) from an annotated corpus and calculates the membership values of the association between an ambiguous word (in a given sense) and the contextual feature, the prior probabilities of the senses, and the likelihood of the contextual feature occurring given the sense. The amalgamation of these elements constitutes the knowledge base on which the subsequent disambiguation phase is to be based.

In order to calculate the aforementioned values, it is necessary to follow the steps described below, which detail the training phase process.

1. For all senses sns_i of the target word w :
   - Use FCM algorithm to calculate the membership values $\mu_{s_{nsi}}(\phi_k)$ for all contextual features ϕ_k.
2. For all contextual features ϕ_k in the corpus:
   - Compute the likelihood:

$$P(\phi_k \mid sns_i) = \frac{\text{Weighted frequency of } \phi_k \text{ in } sns_i}{\text{Total weighted frequency of all features in } sns_i}$$

3. For all senses sns_i of an ambiguous word w :
   - Compute the prior probability:

$$P(sns_i) = \frac{\text{Number of occurrences of } sns_i}{\text{Total number of senses}}$$

The weighted frequency is a frequency measure that incorporates the importance or weight of each instance, as opposed to the more rudimentary approach of raw, unweighted frequency measurement. It provides a methodology for accounting for the relative contribution or significance of individual features in a given context.

The weighted frequency of a feature ϕ_k for a particular sense sns_i is a weighted sum of the contributions of ϕ_k, determined by its membership degree $\mu_{sns_i}(\phi_k)$ to the sense sns_i.

If we consider several occurrences of ϕ_k with different degrees of membership in the sense sns_i, the weighted frequency can be calculated as follows:

$$\text{Weighted frequency of } \phi_k \text{ in } sns_i = \sum_{j=1}^{n} \mu_{sns_i}\left(\phi_k^{(j)}\right)$$

Where:

- n is the total number of occurrences of ϕ_k in the dataset.
- $\mu_{sns_i}\left(\phi_k^{(j)}\right)$ is the membership degree of the j^{th} occurrence of ϕ_k to the sense sns_i.

5.5.2. The Disambiguation Phase: Once the model has been trained, it can now be applied to the disambiguation of a word in a given context on the basis of the FNB expression 9. The disambiguation phase involves using the elements calculated during the learning phase to predict the most likely meaning in a new context.
The following steps must be taken in order to calculate the scores. These steps describe the process of disambiguation.

1. For all senses sns_i of the target word w :
   - Initialize score $(sns_i) = \log P(sns_i)$.
2. For all contextual features ϕ_k in the context window c :
   - Update the score:

$$\text{score}(sns_i) = \text{score}(sns_i) + \log \mu_{sns_i}(\phi_k) + \log P(\phi_k \mid sns_i)$$

3. Select the most appropriate sense sns_{i^*} :
   - Choose sns_{i^*} that maximizes the score:

$$sns_{i^*} = \arg \max_{sns_i \in SNS} \text{score}(sns_i)$$

6. Experiments and Results

6.1. Dataset

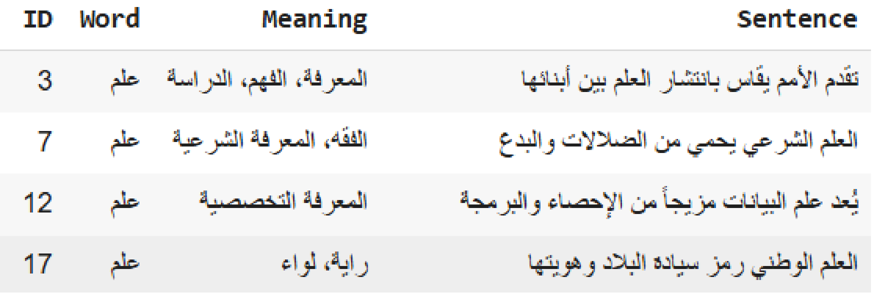
The objective of this study is to evaluate the performance and accuracy of our proposed approach. To achieve this aim, the approach is applied to three separate datasets containing Arabic language data.

- The first dataset under consideration is a manually annotated corpus comprising 40 occurrences of the polysemous Arabic word ”علم“ The sentences were selected from literary and journalistic texts, then annotated according to strict guidelines aimed at identifying the correct meaning of the word based solely on context. The lexical entry is supported by a balanced reference corpus, with each meaning represented by five examples, accompanied by a sentence illustrating the actual usage of the word. This facilitates the study of the disambiguation of the word ”علم“. Figure 2 illustrates an excerpt from the first dataset.

- The second dataset consists of 40 children's stories collected from various online platforms. The text contains 40 ambiguous words, each of which appears in nine different contexts, resulting in a total of 360 annotated sentences. The annotation was conducted manually, employing the Arabic WordNet lexical database to select the most suitable synset for each word, based on its context of appearance. The established guidelines were applied rigorously, thus ensuring consistency in the annotated meanings and avoiding any subjective interpretations not grounded in context.

- The third dataset is the reference corpus proposed by S. Kaddoura et al. [28]. The corpus comprises 3,670 annotated examples covering 367 meanings belonging to 100 polysemous Arabic words. Each meaning is illustrated by 10 contextual occurrences. The corpus was constructed according to a rigorous linguistic protocol, including the following: manual annotation by Arabic-speaking experts; double validation; and systematic arbitration. These measures were implemented to ensure the highest possible quality and inter-annotator consistency. This particular dataset is widely regarded as one of the most comprehensive corpora for lexical disambiguation in Arabic.

For each dataset, a data separation strategy was implemented, utilising an 80/20 division, where 80% of instances were allocated for training and 20% for testing. This division was implemented with the objective of ensuring the complete prevention of data leakage, i.e. ensuring that no information from the test set is utilised during the learning process or parameter tuning. This strategy ensures a reliable and impartial evaluation of the model's performance on data with which it has no prior experience.



ID	Word	Meaning	Sentence
3	علم	المعرفة، الفهم، الدراسة	تقدم الأمم يقاس بانتشار العلم بين أبنائها
7	علم	الفقه، المعرفة الشرعية	العلم الشرعي يحمي من الضلالات والبدع
12	علم	المعرفة التخصصية	يُعد علم البيانات مزيجاً من الإحصاء والبرمجة
17	علم	راية، لواء	العلم الوطني رمز سياده البلاد وهويتها

Figure 2. Extract from the first dataset

6.2. *Experimental Setup*

To evaluate our model, we implemented the steps depicted in Figure 1, and compared it with the Naïve Bayes algorithm based on evaluation metrics [51].
We started with the most important step that directly affects the performance of machine learning methods, which is the preprocessing step.
Table 1 gives an example of the preprocessing steps of the sentence:

" الحرام يبقى حراماً حتى لو كان الجميع يفعله لا تتنازل عن مبادئك دعك منهم فسوف تحاسب وحدك لذا استقم كما أُمرت لا كما رغبت "

(Haram is still haram even if everyone else is doing it, don't compromise your principles and let go of them. You will be held accountable on your own, so do as you are commanded, not as you wish) used in our approach. In the stemming step, several Arabic stemming algorithms were evaluated, including the Farasa [52], Tashaphyne [31], ISRI [32], Snowball [53], and CAMel [33] stemmers.

For POS tagging, we have employed Farasa's tagging algorithm, which is part of the Farasa toolbox, a complete suite for Arabic language processing, and is used to perform tasks such as POS tagging, diacritization, named entity recognition, etc. The utilisation of the POS feature within Arabic WSD systems has been employed by numerous researchers [29], demonstrating its efficacy . The POS of an ambiguous word, as well as the POS of its surrounding words, must be taken into consideration when undertaking sense disambiguation. The performance of the Arabic WSD system is also impacted by the dimensions of the window. In the present experiment, two window sizes were utilised to represent the surrounding word features: 2 and 5 ($S.W._{\pm 2}$ and $S.W._{\pm 5}$). Each of these was associated with its POS properties ($POS_{\pm 2}$ and $POS_{\pm 5}$).

For the word embedding step, the utilisation of a pre-trained model, specifically AraVec [27], is imperative. AraVec signifies a pre-trained set of word embedding models that have been meticulously

Table 1. Example of preprocessing steps.

Methods	Results
Sentence	@<”الحرام يبقى حراماً حتى لو كان الجميع يفعله، لا تتنازل عن مبادئك دعك منهم، فسوف تحاسب وحدك لذا استقم كما أُمرت لا كما رغبت.”>
Normalization	الحرام يبقى حراما حتى لو كان الجميع يفعله لا تتنازل عن مبادئك دعك منهم فسوف تحاسب وحدك لذا استقم كما امرت لا كما رغبت
Tokenization	’الحرام’, ’يبقى’, ’حراما’, ’حتى’, ’لو’, ’كان’, ’الجميع’, ’يفعله’, ’لا’, ’تتنازل’, ’عن’ ’مبادئك’ , ’دعك’, ’منهم’, ’فسوف’, ’تحاسب’, ’وحدك’, ’لذا’, ’استقم’, ’كما’ ’امرت’, ’لا’, ’كما’, ’رغبت’
Stop words removing	’الحرام’, ’يبقى’, ’حراما’, ’الجميع’, ’يفعله’, ’تتنازل’, ’مبادئك’ , ’دعك’, ’تحاسب’ ’وحدك’, ’استقم’, ’امرت’, ’رغبت’
Stemming	’حرم’, ’بقي’, ’جمع’, ’فعل’, ’نزل’, ’بدأ’ , ’ودع’, ’حسب’, ’وحد’, ’قام’, ’امر’ ’رغب’

developed for the Arabic language. The model is based on the Word2Vec algorithm, but adapted to the morphological and syntactic specificities of Arabic (aravec/full grams cbow 300 wikipedia).

The embedding vectors obtained are then sent to the FCM clustering algorithm, which organises the contextual instances into a predefined number of clusters c, corresponding directly to the number of possible meanings of the ambiguous word. The initial search range for the fuzzy coefficient p is set to $[1.1, 5]$, in accordance with recommendations from previous work [60], and its optimal value is determined using the XB index, as demonstrated in Table 2.

The convergence threshold, designated as $\varepsilon = 10^{-5}$, is employed to ensure the precise stabilisation of the membership matrix (for the three datasets). This low value guarantees that the algorithm ceases operation only when successive updates become negligible, thereby circumventing unstable clusters or oscillations in the final iterations. The combination of these parameters is intended to ensure a robust and consistent fuzzy partition.

Upon completion of the optimisation process, the FCM generates a membership matrix, which expresses the degree of association for each instance with each of the possible meanings.

Finally, a Fuzzy Naïve Bayes classifier is formed using the features and membership values to calculate the most probable meaning of the ambiguous word in each sentence.

In this step, we use the smoothing technique [54, 55], which is a method that tries to estimate a probability distribution that is close to the one we expect to find in the stored data. In our case, Arabic WSD, we use smoothing. In order to avoid the effects of zero counts when estimating the conditional probabilities of the model, a very simple smoothing technique proposed by [56, 57] was used in this experiment. It consists of replacing the zero counts of $P(\Phi|SNS)$ by $P(SNS)/n$, where n is the number of training samples.

Fuzziness parameterDataset	First dataset	Second dataset	Third dataset
p = 1.5	0.92	1.04	1.21
p = 2	**0.68**	**0.73**	1.05
p = 2.5	0.79	0.88	**0.69**
p = 3	0.94	0.97	0.83
p = 3.5	1.12	1.15	0.96
p = 4	1.28	1.33	1.15
p = 4.5	1.47	1.52	1.32
p = 5	1.63	1.69	1.48

Table 2. Optimal values of the fuzziness parameter p determined by XB index

6.3. Result and Discussion

In this section, we analyse and discuss the results obtained by applying our Arabic WSD approach to different datasets. This analysis covers several aspects, including the study of cases of incorrect predictions, the impact of the use of different stemming algorithms on the accuracy of the model, and the evolution of the performance according to the parameters and characteristics of the datasets used.

The purpose of this discussion is to highlight the strengths and limitations of the proposed approach and to identify the factors that have a significant impact on its performance.

Table 3 presents a number of examples of incorrect predictions resulting from the application of our lexical disambiguation method. These cases have been chosen to illustrate situations where the proposed approach has difficulty in correctly identifying the meaning of the ambiguous word according to its context.

In the first example, the ambiguous word صاعقة is used in a sentence with a metaphorical register to express a feeling of emotional shock caused by an unexpected departure. The system predicted the meaning خبر صادم (shocking news), whereas the expected meaning was برق قوي (strong lightning). This confusion can be explained by the fact that in a strong emotional context, certain words acquire figurative connotations that can steer the system towards emotional rather than literal meanings. However, in the absence of explicit recognition of the metaphorical register or emotional indicators, the model remains limited to immediate contextual co-occurrence, which encourages this error.

In the second example, the word جرس was predicted in its literal sense of a bell when used as a warning signal (إنذار). Although the context words contain temporal cues and terms related to urgency (وقت العاصفة : time of storm), the semantic similarity between bell and warning was not properly exploited by the model. This highlights a common limitation of systems based solely on contextual feature vectors: the inability to capture the pragmatic and functional relationships between concepts.

Similarly, for the word طبع, the system predicted the meaning of طبع as طبع (character/attitude) when the expected meaning was عدم تأثره بدموع الاخرين (indifference or insensitivity). This error reveals a lack of consideration of causal and emotional relationships in contextual analysis. The sentence appeals to an implicit logic in which an individual's insensitivity to the emotions of others reflects their character, but the system, focusing on direct lexical co-occurrences, was unable to infer this implicit link.

The example of the word عقد, which has several meanings (period, collar, contract...), was misinterpreted in a geopolitical context, where it refers to a decade (عشر سنوات). The system produced scattered predictions, reflecting a structural ambiguity exacerbated by the polysemous richness of this term in Arabic. This highlights the importance of incorporating specialised contextual knowledge or disambiguation models based on enriched semantic networks in such cases.

These incorrect predictions highlight several current limitations of our approach: Difficulty in dealing with figurative and metaphorical expressions, dependence on direct co-occurrences without modelling pragmatic and functional relations, and failure to take into account the typology of the text (emotional, technical, geopolitical, etc.). These observations argue in favour of enriching the model with external semantic resources (such as the Arabic WordNet or conceptual graphs) and integrating mechanisms for register and discourse context recognition in order to improve disambiguation in complex situations.

Table 4 illustrates the effect of using different stemming algorithms on the accuracy of our lexical disambiguation approach. This evaluation was carried out by applying our method to the third dataset, using the following features: surrounding words and POS tags in a window of size ±5 around the ambiguous word to be disambiguated ($S.W._{\pm 5} + POS_{\pm 5}$).

The results clearly show that the Farasa stemmer achieves the best performance with an accuracy of 93.68%, followed by the CAMeL stemmer with 90.54%. On the other hand, the Snowball stemmer recorded the lowest performance with an accuracy of 78.01%. This difference in performance can be attributed to the different morphological processing mechanisms specific to each stemmer. Farasa and CAMeL are tools designed specifically for Arabic and take into account its complex derivational and inflectional morphology. Therefore, these tools are able to preserve the root of words while retaining the contextual information essential for semantic disambiguation. Conversely, more generic stemmers such as Snowball, originally designed for Indo-European languages and later adapted to Arabic, struggle to deal effectively with the morphological peculiarities of this language. This can lead to excessive deletion

Table 3. Examples of a false prediction

Ambiguous word	Sentence	Sentence translation	Predicted sense	Real sense
صاعقة	عندما اكتشف صغار الغزلان أن صديقهم العصفور قد طار بعيدا دون وداع، شعروا وكأن صاعقة من الحزن ضربت غابتهم الصغيرة	When the young deer discovered that their bird friend had flown away without saying goodbye, they felt as if a thunderbolt of sadness had struck their little forest	برق قوي	خبر صادم
جرس	في اللحظة التي رن فيها جرس الإنذار في القرية، فهم الأطفال أن الوقت قد حان للانطلاق بسرعة إلى المخبأ، حيث كانت العاصفة على وشك الوصول	The moment the warning bell rang in the village, the children understood that it was time to quickly head to the shelter, as the storm was about to arrive.	جرس	إنذار
طبع	من. طبع قلبه على والقسوة الأنانية لا تضيره دموع والجوعى المساكين	Those whose hearts are imprinted with cruelty and selfishness are not harmed by the tears of the poor and hungry.	نَسْخ	عوده عليه
عقد	يشار إلى أن نيجيريا تعاني من أسوأ فيضانات منذ عقد ويلقى باللوم في ذلك على فيضان مياه من سد لاغدو في الكاميرون المجاورة إلى جانب هطول الأمطار الغزيرة غير المعتاد	It is noted that Nigeria is suffering from the worst floods in a decade, and the blame is placed on the overflow of water from the Lagdo Dam in neighboring Cameroon, along with the unusually heavy rainfall.	مجموعة من الأشرطة أو الحبال أو الخيوط التي تتشابك بشكل معين لتشكل شكلًا محددًا يستخدم كزينة على الرقبة	فترة زمنية معينة

of suffixes or prefixes, or even to the use of incorrect roots, compromising the quality of the extracted feature vectors and consequently the performance of the disambiguation.

These results confirm the importance of choosing linguistic preprocessing tools adapted to the target language, especially in tasks where the lexical and morphological context is so sensitive, such as lexical disambiguation in Arabic.

Tables 5, 6 and 7 show the performance of our lexical disambiguation approach on the first, second and third datasets as a function of the contextual features used. Two classifiers were evaluated: Naïve Bayes algorithm and Fuzzy Naïve Bayes algorithm. The results highlight not only the importance of the choice of features, but also the crucial role of fuzzy logic in dealing with ambiguous contexts and linguistic uncertainties.

Effect of window size and POS tags: We found that increasing the size of the contextual window around the ambiguous word from ±2 to ±5 significantly improved performance by providing the model with more lexical cues. Adding POS tags to the surrounding words reinforces this trend by integrating an additional morphosyntactic dimension that refines the contextual representation. These results confirm that the richer and more structured the contextual information available to the model, the better its ability to disambiguate. Contribution of fuzzy logic to the Naïve Bayes classifier: The fuzzy logic built into Fuzzy Naïve Bayes plays a crucial role in improving the observed performance. Unlike traditional Naïve Bayes, which is based on strict categorical membership, FNB allows each feature to be assigned a degree of fuzzy membership to different semantic classes. This is particularly relevant for Arabic, where polysemy and contextual variation are common. For example, in an ambiguous context where a word could potentially belong to several senses with close probabilities, Naïve Bayes is forced to choose a single category, increasing the risk of error. FNB, on the other hand, models this uncertainty by assigning confidence levels to each hypothesis and deriving the final class from these fuzzy values. This flexibility allows better handling of cases where the contextual indices are partially contradictory or insufficient.

Table 4. Effect of stemming on the performance of our approach

Stemmer	Precision
Farasa stemmer	93.68 %
ISRI stemmer	88.16 %
CAMel stemmer	90.54 %
Tashaphyne stemmer	82.32 %
Snowball stemmer	78.01 %

The results in Table 5 clearly illustrate this advantage: for all configurations, FNB outperforms NB on all metrics (accuracy, precision, recall, and F1 score). The performance difference is particularly marked in the most rich configurations in contextual information ($S.W._{\pm}5 + POS_{\pm}5$), where fuzzy logic takes full advantage of the density and variability of the context.

The results in Tables 6 and 7, for the second and third datasets, respectively, confirm these observations. Although absolute performance may vary according to the size and nature of the corpora, the consistent superiority of FNB over NB and the beneficial effect of adding POS tags and contextual expansion remain. This shows that fuzzy logic gives the system greater robustness in the face of the diversity of contexts and ambiguities inherent in Arabic.

These results highlight the structuring role of fuzzy logic in the processing of lexical disambiguation. By allowing for degrees of membership and integrating uncertainty into the decision-making process, Fuzzy Naïve Bayes is particularly well suited to the morphological and semantic peculiarities of Arabic. Coupled with extended context extraction enriched with POS tags, this model provides a powerful and flexible solution to dealing with linguistic ambiguity.

Table 5. Results of the proposed approach compared to the NB approach on the first data set.

	NB algorithm				FNB algorithm			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
$S.W._{\pm2}$	72.52%	70.32%	71.13%	70.72%	85.46%	83.48%	84.02%	83.75%
$S.W._{\pm2} + POS_{\pm2}$	78.30%	76.77%	74.00%	75.36%	90.35%	89.08%	88.76%	88.92%
$S.W._{\pm5}$	76.15%	70.89%	68.86%	69.86%	88.90%	86.87%	85.56%	86.21%
$S.W._{\pm5} + POS_{\pm5}$	80.98%	78.46%	78.69%	78.57%	98.00%	95.25%	94.65%	94.95%

Table 6. Results of the proposed approach compared to the NB approach on the second dataset.

	NB algorithm				FNB algorithm			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
$S.W._{\pm2}$	71.50%	79.20%	54.80%	64.60%	80.50%	79.80%	79.10%	79.40%
$S.W._{\pm2} + POS_{\pm2}$	70.30%	78.00%	52.60%	62.60%	83.00%	82.10%	81.50%	81.80%
$S.W._{\pm5}$	72.00%	77.50%	55.10%	64.00%	81.30%	80.70%	80.00%	80.30%
$S.W._{\pm5} + POS_{\pm5}$	73.10%	78.90%	57.80%	66.40%	84.20%	83.50%	82.70%	83.10%

Table 7. Results of the proposed approach compared to the NB approach on the third dataset.

	NB algorithm				FNB algorithm			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
$S.W._{\pm2}$	52.60%	49.80%	44.70%	47.13%	76.56%	85.71%	78.26%	81.52%
$S.W._{\pm2} + POS_{\pm2}$	55.90%	51.70%	46.80%	49.10%	84.88%	87.14%	84.00%	83.43%
$S.W._{\pm5}$	57.30%	53.40%	48.90%	51.00%	84.47%	92.31%	88.89%	90.57%
$S.W._{\pm5} + POS_{\pm5}$	59.00%	55.00%	50.60%	52.70%	83.71%	95.24%	83.33%	88.89%

6.4. Limitations

Although our approach demonstrates strong performance in lexical disambiguation, there are still certain challenges that reflect the task's inherent complexity. For example, figurative or metaphorical expressions and subtle pragmatic relations are difficult to model, and our method currently relies on co-occurrence patterns and contextual features, such as surrounding words and POS tags. Performance can also vary depending on the linguistic pre-processing tools used and the characteristics of the dataset. Addressing these challenges could lead to further enhancements, such as integrating external semantic resources, modelling discourse and register, and enriching contextual representations. This would strengthen the robustness and applicability of the approach.

7. Conclusion and Future Work

WSD has a strategic position in the field of NLP. With the recent emergence of large-scale generative models, such as GPT or LLaMA, capable of producing human-quality texts, the ability to accurately interpret the meaning of words in context remains a fundamental challenge in order to guarantee the coherence and relevance of downstream processing (automatic translation, automatic summarisation, question-answer systems, etc.). From this perspective, effective WSD approaches remain essential to complement and reinforce text generation models, especially in languages with high morphological complexity. In particular, Arabic is characterised by its morphological richness, high polysemy, and the importance of discourse context in the semantic interpretation of words. This complexity makes the task of WSD even more delicate and requires approaches capable of modelling uncertainty and contextual variability flexibly and accurately.

The integration of fuzzy logic through the FNB model proved to be particularly relevant in this work. Unlike traditional approaches based on strict categorical membership, fuzzy logic can handle degrees of membership of multiple semantic classes, which corresponds well to the inherently ambiguous nature of language. This ability to model uncertainty and intermediate contextual situations offers a clear advantage, particularly for Arabic WSD, where many words can simultaneously refer to several similar meanings depending on the immediate context and discourse register. The results obtained on three datasets confirm the robustness and superiority of the fuzzy approach, especially when combined with an extended lexical and morphosyntactic context. This work also shows the growing interest in introducing fuzzy reasoning into other areas of NLP and semantic processing, especially for languages with complex morphology or high contextual load.

For future research, we plan to enrich our approach by integrating deep semantic models, structured lexical resources, and discourse register detection modules to overcome the limitations of the treatment of figurative and metaphorical uses (where the meaning of a word depends on cultural or emotional implicatures). The aim will be to better handle these complex cases and to strengthen the system's ability to disambiguate statements with a high implicit or symbolic content, particularly in literary, journalistic, or religious texts.

References

1. A. Abraham, B. K. Gupta, A. S. Maurya, S. B. Verma, M. Husain, A. Ali, S. Alshmrany, and S. Gupta, *Naïve Bayes Approach for Word Sense Disambiguation System With a Focus on Parts-of-Speech Ambiguity Resolution*, in *IEEE Access*, vol. 12, pp. 126668–126678, 2024.
2. S. Gadri and A. Moussaoui, *Arabic Texts Categorization: Features Selection Based on the Extraction of Words' Roots*, in Computer Science and Its Applications, pp. 167–180, Springer International Publishing, 2015.
3. A. Shimazu and C. A. Le, *High WSD accuracy using Naive Bayesian classifier with rich features*, PhD Thesis, Waseda University, 2005.
4. S. Elmougy, H. Taher, and H. Noaman, *Naïve Bayes classifier for Arabic word sense disambiguation*, in Proceeding of the 6th International Conference on Informatics and Systems, pp. 16–21, 2008.

5. S. Kaddoura, and R. Nassar, *EnhancedBERT: A feature-rich ensemble model for Arabic word sense disambiguation with statistical analysis and optimized data collection*, Journal of King Saud University-Computer and Information Sciences, 36(1), 101911. 2024.
6. M. S. Eid, A. B. Al-Said, N. M. Wanas, M. A. Rashwan, and N. H. Hegazy, *Comparative study of rocchio classifier applied to supervised wsd using arabic lexical samples*, in Proceedings of the tenth conference of language engeneering (SEOLEC'2010), Cairo, Egypt, 2010.
7. L. Merhben, A. Zouaghi, and M. Zrigui, *Lexical disambiguation of Arabic language: an experimental study*, Polibits, no. 46, pp. 49–54, 2012.
8. M. Hadni, S. E. A. Ouatik, and A. Lachkar, *Word sense disambiguation for Arabic text categorization*, Int. Arab J. Inf. Technol., vol. 13, no. 1A, pp. 215–222, 2016.
9. S. Singh, T. J. Siddiqui, and S. K. Sharma, *Naïve Bayes classifier for Hindi word sense disambiguation*, in Proceedings of the 7th ACM India computing conference, pp. 1–8, 2014.
10. K. C. Ryding, *Arabic: A linguistic introduction*, Cambridge University Press, 2014.
11. A. Farghaly, *The Arabic language, Arabic linguistics and Arabic computational linguistics*, Arabic computational linguistics, pp. 43–81, CSLI Publications, 2010.
12. M. P. (Al-Bazi) Khoshaba, *Arabic Grammar Introduced as a Foreign Language: Morphology, Syntax, and Lexicon*, Self-published or unknown publisher, 2007.
13. B. Hammo, A. Moubaiddin, N. Obeid, and A. Tuffaha, *Formal Description of Arabic Syntactic Structure in the Framework of the Government and Binding Theory*, in *Computación y Sistemas*, vol. 18, no. 3, pp. 611–625, 2014.
14. H.-J. Zimmermann, *Fuzzy Set Theory*, in *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 317–332, 2010.
15. G. J. Klir, *Fuzzy Set Theory*, Wiley-IEEE Press, 2006.
16. G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, New Jersey, 1995.
17. L. A. Zadeh, *Fuzzy Sets*, in *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
18. R. Kruse, C. Döring, and M.-J. Lesot, *Fundamentals of Fuzzy Clustering*, in *Advances in Fuzzy Clustering and Its Applications*, John Wiley and Sons, England, pp. 3–30, 2007.
19. P. J. B. Pajila, B. G. Sheena, A. Gayathri, J. Aswini, M. Nalini, and others, *A Comprehensive Survey on Naive Bayes Algorithm: Advantages, Limitations and Applications*, in *Proceedings of the 4th International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, pp. 1228–1234, 2023.
20. S. Ghosh and S. K. Dubey, *Comparative Analysis of K-Means and Fuzzy C-Means Algorithms*, in *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 4, 2013.
21. J. Nayak, B. Naik, and H. S. Behera, *Fuzzy C-Means (FCM) Clustering Algorithm: A Decade Review from 2000 to 2014*, in *Computational Intelligence in Data Mining — Volume 2: Proceedings of the International Conference on CIDM, 20–21 December 2014*, Springer, pp. 133–149, 2014.
22. C. P. Chai, *Comparison of Text Preprocessing Methods*, in *Natural Language Engineering*, vol. 29, no. 3, pp. 509–553, 2023.
23. A. Awajan, *Arabic Text Preprocessing for the Natural Language Processing Applications*, in *Arab Gulf Journal of Scientific Research*, vol. 25, no. 4, pp. 179–189, 2007.
24. S. Salameh, *A Review of Part of Speech Tagger for Arabic Language*, in *International Journal of Computation and Applied Sciences (IJOCAAS)*, vol. 4, no. 3, June 2018.
25. G. Bourahouat, M. Abourezq, and N. Daoudi, *Systematic Review of the Arabic Natural Language Processing: Challenges, Techniques and New Trends*, in *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 3, February 2023.
26. S. Selva Birunda and R. Kanniga Devi, *A Review on Word Embedding Techniques for Text Classification*, in Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020, vol. 638, pp. 267–281, Springer, 2021.
27. A. B. Soliman, K. Eissa, and S. R. El-Beltagy, *AraVec: A Set of Arabic Word Embedding Models for Use in Arabic NLP*, in Proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017), Dubai, United Arab Emirates, Nov. 2017.
28. S. Kaddoura and R. Nassar, *A comprehensive dataset for Arabic word sense disambiguation*, Data in Brief, vol. 55, pp. 110591, 2024.
29. S. Kaddoura and R. D. Ahmed, *A comprehensive review on Arabic word sense disambiguation for natural language processing applications*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 12, no. 4, pp. e1447, 2022.
30. A. G. Al-Khulaidi and S. M. Yaseen, *Comparative Analysis and Evaluation of Stemming and Preprocessing Techniques for Arabic Text*, Sana'a University Journal of Applied Sciences and Technology, vol. 1, no. 4, 2023.
31. R. M. Al-Khatib, T. Zerrouki, M. M. Abu Shquier, and A. Balla, *Tashaphyne0.4: a new Arabic light stemmer based on rhyzome modeling approach*, Information Retrieval Journal, vol. 26, no. 1, pp. 14, 2023.
32. D. H. Abd, W. Khan, K. A. Thamer, and A. J. Hussain, *Arabic light stemmer based on ISRI stemmer*, in Intelligent Computing Theories and Application: 17th International Conference, ICIC 2021, Shenzhen, China, August 12–15, 2021, Proceedings, Part III, pp. 32–45, Springer, 2021.
33. O. Obeid, N. Zalmout, S. Khalifa, D. Taji, M. Oudah, B. Alhafni, G. Inoue, F. Eryani, A. Erdmann, and N. Habash, *CAMeL tools: An open source python toolkit for Arabic natural language processing*, in Proceedings of the twelfth language resources and evaluation conference, pp. 7022–7032, 2020.
34. D. Khurana, A. Koli, K. Khatter, and S. Singh, *Natural language processing: state of the art, current trends and challenges*, Multimedia Tools and Applications, vol. 82, no. 3, pp. 3713–3744, 2023.
35. A. A. Nafea, M. S. Muhmmad, R. R. Majeed, A. Ali, O. M. Bashaddadh, M. A. Khalaf, A. B. N. Sami, and A. Steiti, *A Brief Review on Preprocessing Text in Arabic Language Dataset: Techniques and Challenges*, Babylonian Journal of

Artificial Intelligence, vol. 2024, pp. 46–53, 2024.

36. A. Rayhan, R. Kinzler, and R. Rayhan, *Natural language processing: Transforming how machines understand human language*, in Conference: The development of artificial general intelligence, 2023.
37. M. Bevilacqua, T. Pasini, A. Raganato, and R. Navigli, *Recent trends in word sense disambiguation: A survey*, in International joint conference on artificial intelligence, pp. 4330–4338, 2021.
38. Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., *A survey on evaluation of large language models*, ACM Transactions on Intelligent Systems and Technology, vol. 15, no. 3, pp. 1–45, 2024.
39. S. Kaur, *Advancements in Artificial Intelligence: Machine Learning Techniques and Their Real-World Applications*, Journal of Sustainable Solutions, vol. 1, no. 4, pp. 138–144, 2024.
40. R. Habtamu and B. Gizachew, *State-of-the-Art Approaches to Word Sense Disambiguation: A Multilingual Investigation*, in Pan-African Conference on Artificial Intelligence. PanAfriConAI 2023, vol. 2068, Springer, Cham, 2024.
41. R. M. Nefdt, *The foundations of linguistics: mathematics, models, and structures*, PhD Thesis, University of St Andrews, 2016.
42. L. A. Zadeh, *Probability Measures of Fuzzy Events*, Journal of Mathematical Analysis and Applications, vol. 23, no. 2, pp. 421–427, Aug. 1968.
43. M. Carter and B. Van Brunt, *The Lebesgue-Stieltjes integral*, Springer, 2000.
44. R. M. Moraes and L. S. Machado, *A fuzzy binomial naive bayes classifier for epidemiological data*, in 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 745–750, IEEE, 2016.
45. L. S. Larkey, L. Ballesteros, and M. E. Connell, *Light Stemming for Arabic Information Retrieval*, in Arabic Computational Morphology, vol. 38, pp. 221–243, Springer Netherlands, 2007.
46. E. Mustafa and K. Bouzoubaa, *A Bi-Gram Approach for an Exhaustive Arabic Triliteral Roots Lexicon*, Languages, vol. 8, no. 1, pp. 83, 2023.
47. M. M. Fouad, A. Mahany, and I. Katib, *Masdar: A Novel Sequence-to-Sequence Deep Learning Model for Arabic Stemming*, in Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), pp. 363–373, Springer International Publishing, 2020.
48. T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, 2013.
49. Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, *A neural probabilistic language model*, Journal of Machine Learning Research, vol. 3, no. Feb, pp. 1137–1155, 2003.
50. S. J. Johnson, M. R. Murty, and I. Navakanth, *A detailed review on word embedding techniques with emphasis on word2vec*, Multimedia Tools and Applications, vol. 83, no. 13, pp. 37979–38007, 2024.
51. M. Hossin and M. N. Sulaiman, *A review on evaluation metrics for data classification evaluations*, International Journal of Data Mining & Knowledge Management Process, vol. 5, no. 2, pp. 1, 2015.
52. A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, *Farasa: A fast and furious segmenter for Arabic*, in Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations, pp. 11–16, 2016.
53. J. B. Lovins, *Development of a stemming algorithm*, Mech. Transl. Comput. Linguistics, vol. 11, no. 1-2, pp. 22–31, 1968.
54. S. F. Chen and J. Goodman, *An empirical study of smoothing techniques for language modeling*, Computer Speech & Language, vol. 13, no. 4, pp. 359–394, 1999.
55. S. Aggarwal and D. Kaur, *Enhanced Smoothing Methods Using Naïve Bayes Classifier for Better Spam Classification*, International Journal of Engineering Research and Technology, vol. 2, pp. 3061–3073, 2013.
56. H. T. Ng, *Exemplar-based word sense disambiguation: Some recent improvements*, arXiv preprint cmp-lg/9706010, 1997.
57. G. Escudero, L. Marquez, and G. Rigau, *A comparison between supervised learning algorithms for word sense disambiguation*, arXiv preprint cs/0009022, 2000.
58. S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, *Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence*, Information Fusion, vol. 99, pp. 101805, 2023.
59. B. Taner, and I. B. Türkşen, *Measurement of membership functions: theoretical and empirical work. Fundamentals of fuzzy sets*, Boston, MA: Springer US, 2000. 195-227.
60. Z. KaiLe, C. Fu, and Sh. Yang. *Fuzziness parameter selection in fuzzy c-means: the perspective of cluster validation.* Science China Information Sciences 57.11 (2014): 1-8.
61. X. Zhu, X. Wu, B. Wu and H. Zhou *An improved fuzzy C-means clustering algorithm using Euclidean distance function.* Journal of Intelligent and Fuzzy Systems, 44(6), 9847-9862. 2023.
62. V. Novák. *Fuzzy Sets in Natural Language Processing.* In: Yager, R.R., Zadeh, L.A. (eds) An Introduction to Fuzzy Logic Applications in Intelligent Systems. The Springer International Series in Engineering and Computer Science, vol 165. Springer, Boston, MA. (1992).
63. M. LIU, H. Zhang, Z. Xu, and K. Ding. *The fusion of fuzzy theories and natural language processing: A state-of-the-art survey.* Applied Soft Computing, 2024, vol. 162, p. 111818.
64. C. Gupta, A. Jain, and N. Joshi. *Fuzzy logic in natural language processing–a closer view.* Procedia computer science, 2018, vol. 132, p. 1375-1384.
65. P. Kazemi, H. Karshenas. *Fuzzy word sense induction and disambiguation.* IEEE Transactions on Fuzzy Systems, 2021, vol. 30, no 9, p. 3918-3927.

66. E. VELLDAL. *A fuzzy clustering approach to word sense discrimination.* In : Proceedings of the 7th International conference on Terminology and Knowledge Engineering. 2005. p. 279-292.
67. S. Vij, A. Jain, D. Tayal, and O. Castillo. *Fuzzy logic for inculcating significance of semantic relations in word sense disambiguation using a WordNet graph.* International journal of fuzzy systems, 2018, 20(2), 444-459.
68. A. Jain, D. K. Lobiyal. *Fuzzy Hindi WordNet and word sense disambiguation using fuzzy graph connectivity measures.* ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 2015, 15(2), 1-31.
69. M. N. El-Gedawy. *Using fuzzifiers to solve word sense ambiguation in Arabic language.* International Journal of Computer Applications, 2013, 79(2), 1-8.
70. D. K. Tayal, L. Ahuja, and S. Chhabra. *Word sense disambiguation in Hindi language using hyperspace analogue to language and fuzzy c-means clustering.* In Proceedings of the 12th international conference on natural language processing, 2015, pp. 49-58.
71. I.J. Chiang, C. C. H. Liu,Y. H. Tsai, and A. Kumar. *Discovering latent semantics in web documents using fuzzy clustering.* IEEE Transactions on Fuzzy Systems, 2015, 23(6), 2122-2134.