Educational Administration: Theory and Practice
2024, 30(5), 14971 -14982
ISSN: 2148-2403
https://kuey.net/

Research Article

A Comprehensive Review of Word Sense Disambiguation Research in few Indian Languages: Implications for Educational Tools

Zankhana B. Vaishnav[1*], Priti Srinivas Sajja[2]

[1*]*Sarvajanik College of Engineering & Technology, Sarvajanik University, TIFAC-CORE Building, Dr. R. K. Desai Marg, Athwalines, Surat, 395001, Gujarat, India. zankhana.vaishnav@scet.ac.in*
[2]*Sardar Patel University, Vallabh Vidyanagar, Anand, India. priti@pritisajja.info*

***Corresponding Author:** Zankhana B. Vaishnav
*(9724307552) (ORCID ID: https://orcid.org/0000-0002-5461-7351)

Citation: Zankhana B. Vaishnav, et al (2024) A Comprehensive Review of Word Sense Disambiguation Research in few Indian Languages: Implications for Educational Tools, *Educational Administration: Theory and Practice*, 30(5), 14971-14982
DOI: 10.53555/kuey.v30i5.7968

ARTICLEINFO

ABSTRACT

Natural Languages are inherently ambiguous. Word Sense Disambiguation is one such problem where one word has multiple meaning depending upon the context in which it appears in the text. It can be considered as an intermediate step for many NLP applications like Machine Translation, Summarization, Query Processing, etc. Developing a Word Sense Disambiguation (WSD) system for the Gujarati language can have significant applications in the education sector. There are different approaches available for WSD which includes Supervised, Unsupervised and knowledge-based approaches. The work done is English language is extensive for this problem but for other regional languages more research needs to be done. This paper presents various works, mentioning their proposed method, datasets used, limitations and performance for some Indian languages like Hindi, Malayalam, Bengali, etc. The paper also enlists some general observations about existing approaches for word sense disambiguation. In the last section, the paper proposes a method to resolve WSD problem for Gujarati Language using Genetic Algorithm.

Keywords: Word sense ambiguity; WordNet; Natural Language Processing; Context; Polysemy; Hindi, NLP

1. Introduction

Humans use natural languages for communicating their thoughts to other human beings. Natural language processing is related to human–computer interaction, where several challenges involve natural language understanding. Word sense disambiguation problem (WSD) consists in the computational assignment of a meaning to a word according to a particular context in which it occurs. A word can have number of senses, which is termed as ambiguity. This word sense disambiguation is an intermediate task, but rather is necessary at one level to accomplish most natural language processing tasks. (Navigli & Crisafulli, 2010) (J. Sarmah, 2016). WSD systems integrated into educational instruments promise much in learning and teaching language, particularly in multilingual educational settings. Precise meaning of words based on context using WSD has a great contribution to making the learner's experience more personalized and effective.

2. Applications of Word sense disambiguation in Educational Sector

- **Language Learning Tools**
  - **Vocabulary Enrichment:** WSD can be incorporated into learning tools that expose students to the varying senses of a word depending on the context in which it is used. By learning how words can have different meanings by changes in sentence composition, the student will better appreciate the mother tongue.
  - **Contextual Learning:** The meaning of words may be illustrated by using a disambiguation of words, explaining how, based on proper context, the meanings of words can be understood. This is particularly important in the second-language setting, in which learning students need to appreciate how the context facilitates changes in meaning.

- **Interactive Learning Tools:** The system can be incorporated into educational applications or platforms where the students input sentences and the system displays the possible meanings of ambiguous words. This way, it can enhance comprehension and contextual language learning.

• **Reading Comprehension**

- **Textbooks and Study Material Annotation:** The educational resources contain such intricate texts in which one discovers the ambiguous words to confuse. By applying WSD system, textbooks or study materials can be annotated so that a student feels less confused with the meaning of words related to the particular subject.
- **Helping Tools for Difficult Texts:** The system will give the correct meaning to difficult or complex words in readings, which will enable students to understand literature, science texts, or social studies better in Gujarati.

• **Systems for Questions and Answers for Students**

- **Automated Tutoring Systems:** WSD system will be helpful in developing automated question-answer systems. For example, questions could be generated by students in Gujarati, and system can disambiguate them to answer correctly. Such applications can be of special use in rural or low-resources educational sites where students don't receive much human tutoring due to the limitations.
- **Smart Essay Evaluation:** This system will eventually provide better understanding of the context behind their writings and ensure they use the right words in the right sense.

• **Language Translation Tools for Education**

- **Learning Tools Gujarati to English for students:** Learners of English can use the WSD system when translating from the target language to English. WSD works in finding proper meaning words using context. The system can therefore enhance accuracy of translation tools among Gujarati to English, hence helping their learners to bridge language gaps.
- **Multilingual Education Resources:** WSD can be applied to develop bilingual or multilingual educational resources that assist children in switching between Gujarati and other languages for example, Hindi and English. It is applicable to any multi lingual Indian state where education is conducted in more than one language.

• **Support of Students with Special Needs**

- **Tools for Dyslexic or Learning-Disabled Students:** A WSD system can be used to clean texts by selecting the most plausible sense of ambiguous words and provide those texts in a more accessible format for students with learning disabilities to comprehend.
- **Assistive Technologies:** In addition to speech-to-text, other assistive technologies, WSD system can help students who use text-based aids for the comprehension and acquisition of Gujarati.

• **Support from Teachers in the Classrooms**

- **Development of Assessment:** WSD-based systems may be employed by teachers for more developed assessments of the ability of students to comprehend the context. For instance, an ambiguous sentence may contain ambiguous words that need to be picked up by the students for the right meaning from the context itself.
- **Reducing Ambiguity in Teaching Material:** The WSD system can help teachers eliminate ambiguity in the teaching material provided in a classroom, thereby making the message crystal-clear in subjects like language arts, history, and even science.

• **E-Literature**

- **Enhancing E-Learning Platforms:** The WSD system can be incorporated into e-learning platforms and regional e-books. While students read their digital textbooks, the system will word-inform them in real time, thus helping them understand complicated texts as well as classic literature written in regional languages.
- **Help Promoting Regional Language:** Accessibility of better contextual and meaningful digital resources can also be an opportunity for enhancing the use of regional languages at schools. The system, being designed for both native speaker and language learner populations, can contribute to the development of the use of the regional language at educational levels.

• **Semantic Search in Educational Databases**

- **Intelligent Search Engines for Learning Resources:** WSD can further optimize search features in educational portals. Correct meaning of search phrases in the context will allow students to fetch more relevant study material, references, or research paper from a database.
- **Digital Libraries:** A WSD technique can enable schools and colleges with digital libraries to provide smarter search options. For example, if a student searches for the term "bank," which has multiple meanings, the system will return those results based on context.

3. Hindi Language

In (Tayal, 2015), The training phase starts with processing a set of training documents using the HAL model. The HAL model is used to capture the co-occurrence of words within a given window size in the text. The HAL model generates an $N \times N$ matrix, where N is the total number of unique words in the training documents. It represents the co-occurrence frequencies of words. The HAL matrix is then reduced to remove noise. For each

significant word, a HAL vector is obtained by normalizing the values in the reduced HAL matrix by scaling the values so they sum to a particular value (e.g., 1) or adjusting them to have a mean of zero and a standard deviation of one for easy comparison. The normalized HAL vectors are then clustered using the Fuzzy C-means (FCM) clustering algorithm. Unlike traditional clustering algorithms that assign each data point to a single cluster, FCM assigns a degree of membership to each data point for each cluster. This means that a word can belong to multiple clusters to varying degrees. The result of the FCM algorithm is a set of clusters, where each cluster represents a different context in which an ambiguous word might appear. Euclidean distance between the HAL vector of the target word and the centers of all clusters is calculated and the cluster whose center has the minimum distance to the HAL vector of the target word is identified as a true sense.

Limitations:

- The computational complexity of the approach, especially in the training phase involving HAL vectors and fuzzy clustering, could be a limitation in real-time applications

The authors have used Hierarchical Clustering in (Bhatt, 2015) for Sense disambiguation. After applying pre-processing on the text, context vector is created by taking 2 words left from the target word and 2 words right from the target word. Also, co-occurrence matrix will be created. Then PMI measure is calculated for every Pair of words is context vector. After that using similarity measures like Jaccard, cosine and dice similarity matrix is created which will be then used for hierarchical clustering. Using cluster result as bag of words and synsets[a], glosses, examples and relations from Hindi WordNet, the correct sense for the word is obtained by intersection.

Limitations:

- Research on parts-of-speech other than nouns is lacking, impacting the method's applicability beyond nouns.

The concepts of Fuzzy Hindi WordNet and fuzzy graph connectivity measures is proposed in (Jain A. a., 2015). They developed fuzzy Hindi WordNet which is an extended version of Hindi WordNet. They defined fuzzy relations to show that the concept of composition of fuzzy relations can be used to infer a relation between two words that otherwise are not directly related in Hindi WordNet. The algorithm works in following manner. Construct the sentence graph by performing Depth First Search (DFS) of the Fuzzy Hindi WordNet graph. For the sentence graph, compute all local connectivity measures like degree, eigenvector, closeness, and betweenness centrality for each node. Now for each interpretation of the sentence, they constructed the interpretation graph. The interpretation graph is the subset of the sentence graph. For each disconnected graph a non-zero value of global connectivity measures like compactness, entropy, edge density are obtained. From global and local connectivity, the rank for each content word is calculated and most appropriate sense of the word having highest rank is selected.

In (Athaiya, 2018), the authors have proposed genetic algorithm for Hindi WSD which generates an individual's population by context bag and sense bag. At the initial stage, a population of n chromosomes is created. Each gene of the population is one of the possible senses of the ambiguous word. After determining the fitness of each chromosome, the chromosome with higher fitness value, will help in the reproduction of next generation. Crossover is performed with probability of 0.5 by random selection to form the next stage of generation. The updated population is mutated with mutation probability of 0.15, and new offspring is generated for each chromosome by mutating bits at random positions in the old chromosome.

Limitations:

- The proposed algorithm in the paper only deals with nouns, limiting its applicability to other word types.

In (Jain & Lobiyal, 2019), The input to the algorithm consists of words from a given sentence, which are represented as candidate vertices of a graph. Each word will be a node in the graph. The next step involves extracting the different senses or meanings of the words from the Wordnet and representing them as nodes in the graph. This step forms the basis for disambiguating the senses of ambiguous words. To determine the similarity between various word senses, the Adapted Lesk approach is used. This method calculates the similarity index between different senses of words. An edge is created between two-word senses in the graph, and a weight is assigned based on the similarity index calculated using the Adapted Lesk approach. This weight signifies the strength of the relationship between the senses. After assigning weights and creating edges, the algorithm calculates the indegree of all nodes in the graph. The indegree represents the number of edges pointing towards a particular node. The node with the maximum indegree for each word is selected as the disambiguated sense for that word.

Limitations:

- The proposed system relies on assigning membership values to semantic relations, which may introduce subjectivity and potential biases in the disambiguation process.

[a] A synset, short for "synonym set," refers to a group of words that are synonymous or closely related in meaning. In lexical databases such as WordNet, synsets are organized hierarchically and provide structured information about word meanings and relationships between words.

In (Soni, Gopalaniî, & Govil, 2021), the proposed approach is developed using word vectors from Continuous Bag of Words(CBOW)[b] and Skipgram[c] models of Word2Vec. The approach involves splitting input sentences into sets of ambiguous and unambiguous words, processing ambiguous words through IndoWordNet to collect senses, and using context words to determine the exact interpretation of ambiguous words. The context vectors are created for each ambiguous word, and the best interpretation is determined based on cosine distance from reference vectors.

Limitations:

- Lemmatization of words is not performed before training the model, which could potentially enhance the results.
- Disambiguating words used in idioms poses a challenge as they may have completely different interpretations compared to their present context.

In (Kulkarni & Rodd, 2022), proposed method uses Fuzzified Semantic Relations and Fuzzy Hindi WordNet to clarify word meanings. The network organizes words into synsets, representing semantic relationships like synonyms, hypernyms, hyponyms, and meronyms. Fuzzy logic captures uncertainty and imprecision in word meanings, leading to a more sophisticated understanding of language. Fuzzy sets represent semantic relations between words, accommodating varying degrees of membership. The context of a target word is depicted by fuzzy semantic connections with other words in the text. Fuzzy inference is used to identify the most suitable meaning for the target word, using fuzzy rules and membership functions. The meaning with the strongest association or membership is chosen as the disambiguated meaning for the target word.

In (Bhatia, Kumar, & Khan, 2022), The system is given a text document as input and removes unwanted terms like stop words. Words with multiple meanings are identified as ambiguous and processed through WordNet. Words with multiple meanings are ambiguous, and the rest are neighbors. The neighboring words and the ambiguous word are run through the Hindi WordNet dictionary to extract all of their meanings and two bags called sense bag and context bags are created. Finally, a genetic algorithm is used to determine the correct meaning of these ambiguous words.

In (Yusuf, Surana, & Sharma, 2022) , have developed a pipeline for Hinglish to Hindi transliteration, spell correction, POS tagging, and word sense disambiguation of Hindi text. To determine the correct meaning, the algorithm compared the input sentence with the keywords in the helper dataset to identify which meaning of a particular word had the highest overlap. This overlap comparison was then used to select the most suitable meaning for the ambiguous word as in LESK algorithm.

Limitations:

- Limited resources led to focusing on common words and dictionary-based methods for WSD.
- The dataset for enhancing the Lesk algorithm only included two meanings per word due to time constraints and limited manpower.
- The WSD model may face challenges in understanding complex linguistic and context in Hindi.

In (Padwad, et al., 2024), proposed method first generates Synonym. This process involves calculating the cosine similarity between the first example sentence of each synset and the input sentence. After calculating the cosine similarity, 7 best senses based on the highest similarity scores is considered as the synonym set for the ambiguous. Then, a list of sentences where ambiguous words are substituted with each word from the synonym set is created. After generating the sense-replaced sentences list, the sentences with most similarity to the input sentence are filtered. These sentences serve as input to MuRIL, allowing the model to assess the contextual fit of the potentially disambiguated word within the sentence leading to ambiguity resolution.

Limitations:

- Heavily relies on the availability of a large labeled corpus.
- The algorithm's sensitivity to the precise wording of dictionary definitions can significantly influence the outcomes.

Table 1: Summary of work done on Hindi language.

Authors & Year	Approach used	Dataset & Accuracy
(Tayal, 2015)	Semantic space model HAL and Fuzzy C-means clustering for representing context.	Accuracy – 79.16%
(Bhatt, 2015)	Hierarchical clustering and Hindi WordNet as Knowledge base.	246 words - History, 1279 words - social study and 2672 words - short story. Accuracy - 81.64% - Cosine, 80.38% - Jaccard and 79.74% - Dice similarity for History domain. Precision varies from 74% to 82%.

[b] Continuous Bag-of-Words (CBOW) is a neural network architecture commonly used in natural language processing tasks. In CBOW, the model predicts a target word based on the context provided by the surrounding words within a fixed window size.

[c] Unlike the Continuous Bag-of-Words (CBOW) model, the Skip-gram model predicts the context words based on a given target word.

(Jain A. a., 2015)	Representation of Fuzzy Hindi WordNet using fuzzy graphs and graph connectivity measures (both local and global) for WSD	publicly available sense marked corpus on the IIT Bombay Web site (Centre for English Language Technology 2010). Performance increases in most cases by approx. 8%.
(Athaiya, 2018)	Genetic Algorithm and Hindi WordNet as Knowledge Base	Dataset from TDIL, small testing manually created dataset from the domains like sports, history literature etc. Accuracy of 85-90% for different domains.
(Jain & Lobiyal, 2019)	Extended FHWN with fuzzy semantic relations and centrality measures. Proposed a graph-based approach which Assigns weights to semantic relation edges in Fuzzy Hindi WordNet.	health dataset of IIT Bombay sense tagged corpus. The results are compared with their previous work (Jain A. a., 2015) on the same data set and this approach performance better.
(Soni, Gopalaniî, & Govil, 2021)	CBOW and Skipgram models of Word2Vec	An adaptive approach for WSD in Hindi language was evaluated on a large-scale Hindi corpus, showing better results than previous attempts
(Kulkarni & Rodd, 2022)	HindiSentiWordNet with graph-based Lesk approach for word sense disambiguation.	dataset was developed with 4028 sentences in Hindi from sources such as movie reviews, product reviews and travel reviews
(Bhatia, Kumar, & Khan, 2022)	Genetic algorithm. And wordnet to identify ambiguous word and context representation.	Manually created testing dataset from standard online essays, news, history. Improved Accuracy by 8% compared to existing approaches
(Yusuf, Surana, & Sharma, 2022)	Custom LESK Algorithm	A dataset of 20 ambiguous Hindi words was curated. Accuracy - 71%

4. Malayalam

In (Gopal, 2016, March), authors have used Naïve Bayes classifier for WSD task for Malayalam language. After all the preprocessing phase which consists of stemming, tokenizing and stop word removing. Then the conditional probabilities of the various senses of an ambiguous word, relative to the feature vectors, are calculated using a sense corpus and applying the Naïve Bayes classifier. They have used nouns as features in this method.

Limitations:

- The quality of the Word Sense Disambiguation (WSD) system is directly proportional to the quality of the corpora employed in the system.

In (Junaida, 2017, December), the authors have suggested two algorithms Conditional Random Field (CRF) and Margin Infused Relaxed (MIRA) in a CRF framework for Malayalam WSD. A classifier predicts a label for a single sample without considering neighboring samples; a CRF can take context into account. For context representation two groups of features were used, Word and Word + Part-of-speech bigrams. Word features are lexical features; unique words that occur in the training set in a specific window range Word + POS features are lexico-syntactic features combining POS information in a predefined range of the particular word. For these experiments, the method of 10-fold cross validation is used divided in ten sets, each set containing 10% of the total data.

Limitations:

- The evaluation metrics used for the CRF and MIRA models may not fully reflect the classifier's performance due to the highly imbalanced data, leading to potential inaccuracies in the assessment.
- The average F-measure obtained from the 10-fold cross-validation using CRF for word and word POS features was 52.35, indicating room for improvement in the model's overall performance.

Table 3: Summary of work done on Malayalam language.

Authors & Year	Approach used	Dataset & Performance
(Gopal, 2016, March)	Naïve - Bayes classifier	Ambiguous corpus: This corpus contains all the ambiguous words in Malayalam language and Sense corpus - Accuracy – 90%
(Junaida, 2017, December)	Conditional Random Field (CRF) and Margin Infused Relaxed (MIRA) in a CRF	Manually collected sentence from various Malayalam newspapers, Wikipedia articles, blogs, books, novels etc. Using different combinations of CRF, MIRS, word feature and POS feature, they have achieved average - Precision/Recall/F-measure - 61.64/57.64/55.79

5. Assamese

In (Kalita, 2015, September), authors have proposed WSD system based on Walker algorithm. They have prepared a text file that has sample words, their subject category and domain. One CONTEXTBAG is created which has all the subject categories of the context words in the sentence. The sense category with maximum matches gives the sense of ambiguous word.

Limitations:

- The system relies on a modified version of the Assamese wordnet due to the unavailability of an existing thesaurus with tagged categories for the Assamese language.
- The algorithm's performance can vary based on the word window parameter, which determines the number of neighboring words considered for disambiguation. A smaller word window may lead to lower recall values.

In (Sarmah, 2016) , authors have proposed a supervised WSD system based on decision tree. Duplicate words filtration was done to get a wordlist from the 50K sentences. Words which occur in different Synset entries are extracted first and manually validated to derive ambiguous words from WordNet. The system consists of the modules: pre-processing raw data, sense inventory preparation, feature/attribute selection, preparing the decision tree. After pre-processing, for local lexical feature selection left and right features of the target word with the range {-2, -1, 0, +1, +2} are taken. Decision tree is created based on the features extracted in the previous step. Two types of evaluation procedures are performed. First, hold out evaluation splits the sense-annotated data ensuring that each class is represented in both training set and test set and second, k-fold cross-validation to improve the performance.

Limitations:

- Complex machine learning algorithms like decision trees may lack interpretability, making it hard to understand the model's decision-making process.

In (Borah, 2019), the authors have used Naïve Bayes classifier for WSD. Naïve Bayes classifier works on the assumption that all the features, which are used to classify the test case are conditionally independent. Using this, authors have found out the sense which maximize the conditional probability. They have used five different features namely Unigram Co-occurrence (UCO), Parts of Speech of Target word (POST), Parts of Speech of Next word (POSN) and Local Collocation (LC) and Semantically Related Words (SRW). The basic drawback of this system is the size of the training corpus and test corpus due to which the current system is suitable for a selected set of nouns, adjectives, verbs, pronouns and quantifiers with less effects of morphology.

Limitations:

- While the addition of the Semantically Related Words (SRW) feature improved the system's performance, the effectiveness of this feature may vary depending on the specific characteristics of the ambiguous words being disambiguated.

In (Gogoi, Baruah, & Sarma, Assamese Word Sense Disambiguation using Genetic Algorithm, 2020), the authors have proposed genetic algorithm for disambiguation task. Each chromosome in a given population represents all possible senses for a specific term in WordNet. Then Wu-Palmer's similarity measure is employed as the fitness function. This measure calculates the similarity between two senses based on their depth in the WordNet hierarchy, aiding in determining the most suitable sense for a given term. The sense that scores the maximum similarity using the fitness function is declared as the winner sense.

Limitations:

- The proposed system faces challenges with short sentences, spelling errors, lack of information in Assamese Word-Net, and different senses for the same contextual words, impacting data retrieval and similarity measurement

In (Gogoi, Baruah, & Nath, Assamese Word Sense Disambiguation using Cuckoo Search Algorithm, 2021), the authors have proposed cuckoo search algorithm for WSD. The Assamese wordnet is used to collect all the possible senses for population initialization. To check the fitness of each cuckoo they have used Cosine's similarity as the fitness function. Fitness function is used to summarize, as a single figure of merit, to know how close a candidate solution is to achieving the set aims. Cosine similarity measures similarity between vectors in an inner product space, determining if they point in the same direction. It provides a quantitative measure of similarity based on the cosine of the angle between vectors, indicating higher similarity with smaller angles.

Limitations:

- Lack of comparison with more diverse or state-of-the-art algorithms in the field
- Limited discussion on the scalability of the Cuckoo Search Algorithm for larger datasets
- The focus on the Assamese language may limit the generalizability of the findings to other languages

Table 4: Summary of work done on Assamese language.

Authors & Year	Approach used	Dataset & Performance
(Kalita, 2015, September)	Walker's Algorithm and modified Assamese WordNet XML file with word Category	Random sentences from the Internet. Precision - 86.66, Recall - 61.09.
(Sarmah, 2016)	Decision tree algorithm	160 ambiguous words - corpus, 100 ambiguous words - WordNet, 50k sentences are tagged with the appropriate sense manually. Average F-measure - 0.611 (10-fold cross validation evaluation on 10 words.)
(Borah, 2019)	Naïve Bayes classifier	External knowledge source for training and testing phase. Accuracy - 56.2 to 91.11%. Highest F1-measure of 91.11%
(Gogoi, Baruah, & Sarma, Assamese Word Sense Disambiguation using Genetic Algorithm, 2020)	Genetic Algorithm	Sense-annotated Assamese corpus derived from Assamese WordNet, consisting of 50,001 words, with 15,606 ambiguous nouns. Precision - 81.25, Recall - 74.28
(Gogoi, Baruah, & Nath, Assamese Word Sense Disambiguation using Cuckoo Search Algorithm, 2021)	Cuckoo Search Algorithm	An Assamese corpus with senses from the WordNet, with 50,000 words, of which 15,606 were ambiguous. Precision of 87.5%, Recall of 84%, and F-measure of 85.71%

6. Bengali

In (Pal A. R., 2017, February), the authors have proposed Type-based and Token-based clustering strategies for sentence clustering. They have done text normalization, text lemmatization and stop word removal from the test corpus. For feature selection, they calculated term frequencies of the individual keywords present in the document and after pruning the least occurring words, the remaining words have been selected for a feature vector. For types-based clustering, they prepared matrix of vectors of sentence and used K-means algorithm (k =2 was taken for experiments) to cluster overall test data. For token-based clustering, the meanings of the test sentences were expanded using the Bengali WordNet. This extended sense definitions of the sentences are clustered using the same algorithm. For sense assignment, a specific sense is assigned to the clusters according to the meanings of the member sentences of the clusters by a linguistic expert. These clusters are used as a referenced sense repository. New test sentence tagged with that particular sense of the cluster, to which it is closest.

Limitations:

- The clustering process in the system involves manual labeling of clusters by a linguistic expert, which can be time-consuming and subjective.
- The text normalization process in the system involves manual procedures to remove punctuation symbols and normalize the text.

In (Pal A. R., 2017), first the input sentences have been retrieved randomly from the Bengali Text Corpus. The text retrieved from the Bengali corpus was has been normalized manually. Next, the normalized text is lemmatized. After removing the stop words from each sentence the meaningful words and the synonymous words of the meaningful words have been accumulated with the help of the Bengali WordNet. Next, the glosses and example sentence of each of these words have been retrieved from the WordNet and concatenated to form a string. Then, the individual sense-carrying gloss of the ambiguous word has been compared with the string to find the overlap. The maximum overlap resolute the actual sense of the ambiguous word in that particular context.

Limitations:

- The approach's effectiveness may vary depending on the complexity and ambiguity of the words being analyzed

In (Pandit, 2018, January), the authors have proposed three modified LESK algorithms for WSD of Bengali words. In first version, a glossbag is created which is an array that contains the collection of senses of a particular word which is derived from the WordNet and overlap is calculated between glossbags of every senses. In distance based LESK, a priority is assigned to each context word based on its distance from the target word and overlap is calculated taking distance into account. The intuition is that words that lie further away from the target word contribute less in disambiguating the target word. In dependency tree based LESK, they used pair of complementary information like distance between the context word and the target word in the dependency tree, and semantic similarity between the context word and the target word. Semantic similarity

rewards those words that lie further from the target word in the dependency tree, but are highly related. They modified overlap formula incorporating distance and semantic similarity.

Limitations:

- The paper does not delve into the computational complexity or efficiency of the proposed algorithms, which could be crucial for real-time applications or large-scale text processing tasks.

In (Sau, Amin, Barman, & Pal, 2019) the authors have proposed Bengali Word Sense Disambiguation using Sense Induction technique. They used distributional method to analyze the distributional properties of words in large text corpora to identify clusters of words that co-occur in similar contexts. Words within the same cluster are assumed to share a common sense. For semantic similarity they expanded the context by considering synonyms and gloss of every Content word of the sentences. They also Addressed challenges in sense classification for ambiguous words and provided insights for developing effective algorithms in language processing.

In (Saha, Das Mou, & Mittra, 2019) the authors have proposed Bangla WSD system using Levenshtein Distance and Cosine Similarity. They have used Levenshtein edit distance algorithm to detect ambiguous word and the feature word from the text. The word is selected as ambiguous word if it has minimum distance and next word of ambiguous word in given sentence is selected as feature word. Then system begins by trying to find a match for the keyword in the text collection. If a match is found, it displays the corresponding meaning. If no match is found, the system calculates the similarity between the input sentence and all sentences in the collection. The highest similarity score is then compared to a threshold of 0.6. After testing. If the similarity score exceeds the threshold, the system predicts the meaning of the input sentence using the matched sentence from the collection.

Limitations:

- The method relies on the Levenshtein distance algorithm and Cosine Similarity for word sense disambiguation, which may not be the most advanced or optimized techniques.

In (Das Dawn, Khan, Shaikh, & Pal, 2024), there are five key elements for disambiguation task. After the preprocessing task, a set of features is gathered from the text data. They have used Local features like density of keyword in the document and global features like rare terms, keywords of sense of ambiguous word. The features are then extracted using the CLARF score of each lexeme. CLARF (cohesive lexical ambiguity revealing factor) helps in quantifying the syntactic relationships of lexemes in the text. the next step involves calculating the integrated lexeme connexion measure from the testing set to the training set. This measure aids in understanding the connectivity and relationships between different lexemes. Finally, the testing data is recognized by applying the max-rule of the LCM (lexeme connexion measure) score. The algorithm considers various factors like frame lexeme harmony, sense lexeme harmony, and polysemy distribution, enhancing the accuracy of sense recognition.

Limitations:

- The system struggles with identifying adverbial nouns and compound words in Bengali text.

Table 5: Summary of work done on Bengali language

Authors & Year	Approach used	Dataset & Performance
(Pal A. R., 2017, February)	Type-based and Token-based clustering using K-means algorithm.	1371 sentences of a Bengali ambiguous word (Ghantā). Baseline accuracy - 54%, Accuracy of clustering of the sentences - 63% (after sense expansion)
(Pal A. R., 2017)	Overlap approach and Bengali WordNet as Knowledge base.	485 sentences of 9 ambiguous words. P=R=365/485=0.75 and F-Measure= 0.75
(Pandit, 2018, January)	Three different Modified LESKs using gloss overlap, distance and semantic similarity. The Bengali WordNet is used as the lexical database.	22 words having 91 senses. 11 nouns, 5 verbs and 6 adjectives. Test set 1 - 75 manually constructed sentences. Test set 2 - 79 example sentences from the Bengali WordNet
(Sau, Amin, Barman, & Pal, 2019)	Sense Induction technique by measuring Semantic Similarity. Context Expansion using WordNet.	10 commonly used Bengali ambiguous words. Those are- (hāt), (māthā), (fal), (man), (kathā), (kapāl), (nāk), (jal), (mānus) and (pā). Accuracy - 63.71%
(Saha, Das Mou, & Mittra, 2019)	Utilizes Levenshtein Distance algorithm and Cosine Similarity for detection.	created a training corpus of 3860 sentences by collecting sample data from Bangla grammar books, newspapers,blogs, an open-source Bangla corpus and informal survey. Accuracy – 80.82%
(Das Dawn, Khan, Shaikh, & Pal, 2024)	Lexeme Connexion Measure of Cohesive Lexical Ambiguity Revealing Factor	The proposed algorithm's performance was evaluated on a dataset consisting of 100 polysemous words with three/four senses

7. Observations:

We have reviewed available research papers for some Indian languages. Limited work has been done in other Indian languages also. We have observed following points from the reviewed work:

- For approaches which use overlap approach, the Indian languages are vast in semantic nature and the supporting data sets to represent a particular sense are so different that overlap cannot occur in few cases.
- The vast semantic varieties in sentences sometimes made it impossible to programmatically track.
- The scarcity of data in the IndoWordNet is a big issue. As that is still under development, it is not a complete reference of knowledge base in Indian languages.
- There are very large sentences with many irrelevant contextual words and very short sentences with lack of sufficient information within them, feature selection or context representation can be a difficult task.
- Some sentences exhibited similar senses without contextual word similarities, making it challenging for automated systems to disambiguate accurately
- In AI Methods, some ideas form the basis of all further work on the subject and its very domain specific.
- In Knowledge based approaches, Accuracy Rely on precompiled lexical knowledge resources.
- In Supervised approaches Accuracy depends on pre-annotate corpora for training data which is very scarce for low resource Indian languages.
- For unsupervised approaches, no pre-training necessary and can work on multiple languages with no modification to the algorithm.
- There is some work done for Indian languages like, Telugu, Punjabi, Urdu, Marathi but no works has been done in Gujarati language.
- Most of the work done for Indian languages use knowledge-based approaches for the WSD task.
- IndoWordNet developed by IIT Bombay mostly used as knowledge resource.
- Most of the authors have used Precision, Recall, F-Measure, and Accuracy as evaluation measures. Some have used K-Fold cross validation.

8. Proposed method

We propose a method to solve the problem using knowledge-based approach which uses Gujarati Wordnet (Bhadeliya, & Joshi , 2018) as lexical resource. This approach takes multiple sense of word (meanings) from WordNet and generate context from the input text. This information is then used by the fitness function of genetic algorithm. Fitness function calculates fitness of the candidate solution using context information and fuzzy logic to tag words in input sentences with the best suitable sense.

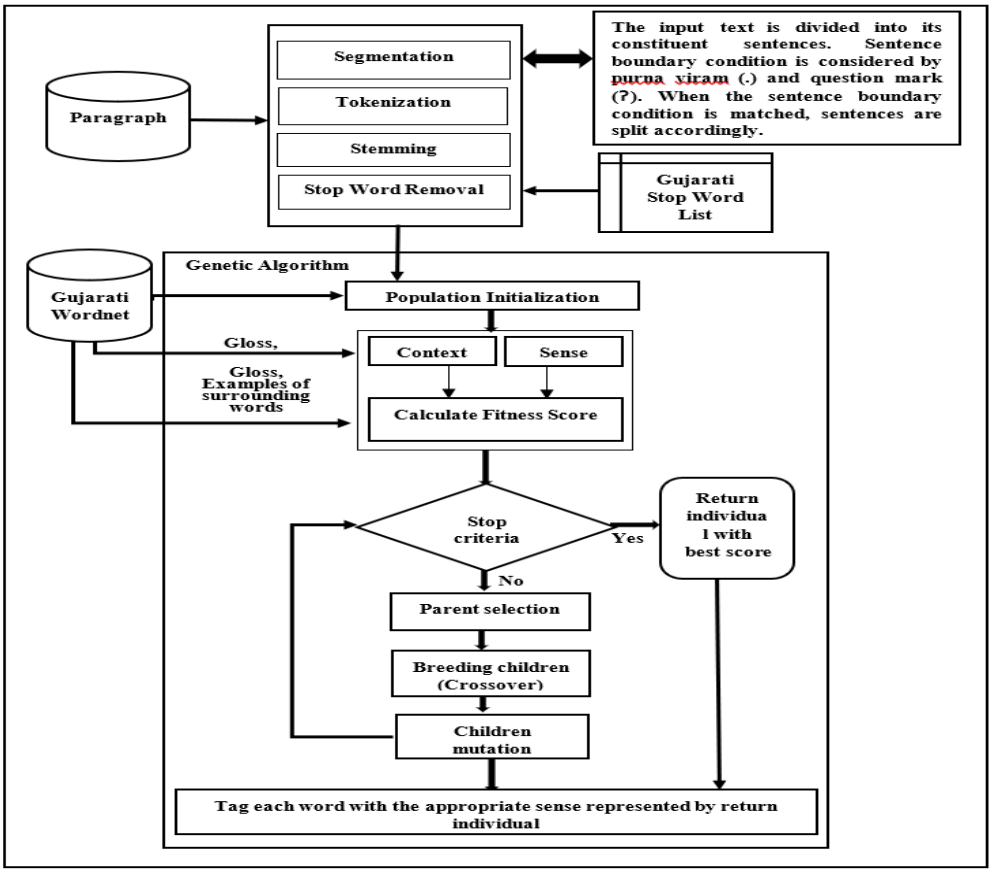

Fig. 2 Overall structure of proposed system

8.1. Preprocessing phase

In this phase we will clean and transform the text to be used for further work.

- **Sentence segmentation**

In sentence segmentation, sentence boundary condition in the Gujarati language is considered by purna viram (.) and question mark (?). We will break down a given paragraph into individual sentences.

- **Tokenization**

This process takes the output from sentence segmentation as an input, and then sentences are split into tokens (words) when special symbols like commas, space, etc., are coming in between words (S. S. Pandya, 2021).

- **Stop word removal**

Stop words are the most frequently used words like articles, operative words, prepositions, conjunction, etc. Stop words do not carry any relevant information, so they should be eliminated from the input text.

8.2. Genetic Algorithm

Motivated by the strength of the population-based algorithms, we propose the use of Genetic algorithm for the WSD task.

8.2.1. Population Initialization

```
Input: W1, W2, W3, ...., Wn  /*n is the Text size */
IndividualList <- empty list of inviduals
W = MaxSenses{W1, W2, W3, ...., Wn} /* word with the maximum count */
popSize = W.SenseCount
for (i =1 to popSize) do
    for (j= 1 to n) do
        if i mod SenseCount(Wj) = 0 then
            individual = concate(individual , SenseCount(Wj))
        else
            individual = concate(individual , i mod SenseCount(Wj))
    end
    IndivudualListi = individual
end
Output: IndividualList
```

Fig. 1 Population Initialization Algorithm

For every input paragraph, population size and length of an individual will be variable. Chromosome length will be the maximum of words sense count in input string and each chromosome has genes equal to the number of words in input string. Using following algorithm we can generate initial population.

8.2.2. Fitness function calculation

To calculate fitness score we are using two parameters.

1. Overlap between the meaning and example of targeted word and its surrounding words. We will use Indo-Aryan Wordnet to fetch meanings and examples of Gujarati words in a text.
2. Matching of “Part of Speech” for each word in candidate solution and targeted text.

It is difficult to decide the precise value of these two features as it depends on the inputted text and surrounding context words. Due to this uncertainty we will use fuzzy logic controller(FLC) to mimic human reasoning and decision making.

Following are the key concepts of fuzzy logic controller:

Fuzzy Logic: Unlike classical logic where variables are either true or false (0 or 1), fuzzy logic allows for degrees of truth. Variables can take any value between 0 and 1, representing the degree to which a statement is true.

Fuzzy Sets: In fuzzy logic, variables are described by fuzzy sets. A fuzzy set is defined by a membership function, which assigns a degree of membership (between 0 and 1) to each element in the set.

Linguistic Variables: Variables are often expressed in linguistic terms, such as "low," "medium," and "high," rather than precise numerical values.

Components of a Fuzzy Logic Controller:
Fuzzification: Converts crisp input values into fuzzy values. This involves mapping numerical input data to fuzzy sets using membership functions. For example, if the input is temperature, it could be fuzzified into categories like "cold," "warm," and "hot."
Rule Base: A collection of fuzzy if-then rules that describe the desired behavior of the system. Each rule defines an action to be taken for a given set of fuzzy input conditions in IF…THEN rule.
Inference Engine: Processes the fuzzy inputs according to the rules in the rule base to generate fuzzy outputs. It determines the degree to which each rule applies to the current situation and combines the results to produce a set of fuzzy output values.
Defuzzification: Converts the fuzzy output values back into crisp values that can be used to control the system.

In our algorithm, input variables are overlap and POS matching, output variable is final fitness score. Using fuzzification we will convert their crips values to fuzzy values like 'Low', 'Medium' and 'high' using triangular membership function. Then we will define rule base. This rule takes form of "IF…THEN…". for example,

IF score1 is 'low' and score2 is 'low' THEN final_score is 'low'.
Here score1 and score2 are input variables and "&" is "logical AND" operation. The final_score is output variable.
These rules are then combined into a fuzzy system, and a simulation instance is created to apply these rules to specific input values.

8.2.3. Parent selection, Crossover, Stopping Criteria

After calculating fitness of all individuals in population, parent selection is done for mating to generate next population. We propose fitness-proportionate selection, also known as roulette wheel selection. In this method, individuals are selected according to their fitness scores, such that individuals with higher fitness scores have a higher probability of being selected.
Crossover operation generate new individual from the selected parents. We can use single point crossover to generate new individual.
The algorithm will stop either when the generation reaches to predefined number or the best solution during the evolution process doesn't change to a better value for a predefined value of generations. Finally the algorithm will tag sense to each word in a text.

9. Conclusion

This paper focused on the work done in the area of word sense disambiguation (WSD) for few Indian languages (S. Mulkalapalli, 2016). Out of many Indian languages most of the work is done in Hindi language. The main complication is the lack of required corpus resources for developing WSD system. For languages like Gujarati, the availability of required resources for developing WSD system is less to our knowledge. In the future, we would like to develop a comprehensive WSD system for Gujarati using populations-based algorithm like genetic algorithm and expand it as Multilingual System for WSD.

Data availability statement: This is a review article. So data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

1. Athaiya, A. D. (2018). A Genetic Algorithm Based Approach for Hindi Word Sense Disambiguation. *Proceedings of the 3rd International Conference on Communication and Electronics Systems (ICCES). IEEE, 2018.*, (pp. 11-14). Coimbatore.
2. Bhadeliya,, D., & Joshi , H. (2018). Gujarati WordNet: Development, Expansion and Usage. *Proceedings of the 11th Global WordNet Conference*.
3. Bhatia, S., Kumar, A., & Khan, M. (2022). Role of Genetic Algorithm in Optimization of Hindi Word Sense Disambiguation. *IEEE Access, 10*, 75693-75707.
4. Bhatt, N. P. (2015). Hierarchical clustering technique for word sense disambiguation using Hindi WordNet. *2015 5th Nirma University International Conference on Engineering (NUiCONE)*, (pp. 1-5). Ahmedabad.
5. Borah, P. P. (2019). WSD for Assamese Language. In J. B. Kalita (Ed.), *Recent Developments in Machine Learning and Data Analytics* (pp. 119-128). Singapore: Springer.
6. Das Dawn, D., Khan, A., Shaikh, S., & Pal, R. (2024). Lexeme connexion measure of cohesive lexical ambiguity revealing factor: a robust approach for word sense disambiguation of Bengali text. *Multimedia Tools and Applications, 83*(5), 12939–12983.
7. Gogoi, A., Baruah, N., & Nath, L. (2021). Assamese Word Sense Disambiguation using Cuckoo Search Algorithm. *International Conference on AI in Computational Linguistic*, *189*.

8. Gogoi, A., Baruah, N., & Sarma, S. (2020). Assamese Word Sense Disambiguation using Genetic Algorithm. *Proceedings of the 17th International Conference on Natural Language Processing (ICON).*
9. Gopal, S. &. (2016, March). Malayalam word sense disambiguation using Naïve Bayes classifier. *2016 International Conference on Advances in Human Machine Interaction (HMI) IEEE.*, (pp. 1-4). Doddaallapur, Bangalore.
10. J. Sarmah, S. S. (2016, May). Survey on Word Sense Disambiguation: An Initiative towards an Indo-Aryan Language. *I.J. Engineering and Manufacturing, 6*(3), 37-52. doi:DOI: 10.5815/ijem.2016.03.04
11. Jain, A. a. (2015, December). Fuzzy Hindi WordNet and Word Sense Disambiguation Using Fuzzy Graph Connectivity Measures. *ACM Trans. Asian Low-Resour. Lang. Inf. Process., 15*(2), 8:1-8:31.
12. Jain, G., & Lobiyal, D. (2019, 1). Word Sense Disambiguation of Hindi Text using Fuzzified Semantic Relations and Fuzzy Hindi WordNet. *9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 494-497). IEEE.
13. Junaida, M. K. (2017, December). Word Sense Disambiguation for Malayalam in a Conditional Random Field Framework. . *In Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, (pp. 495-502). Kolkata.
14. Kalita, P. &. (2015, September). Implementation of Walker algorithm in Word Sense disambiguation for assamese language. *In 2015 International Symposium on Advanced Computing and Communication (ISACC). IEEE.*, (pp. 136-140). Silchar, India.
15. Kulkarni, D., & Rodd, D. (2022, 1). Word Sense Disambiguation for Lexicon-based Sentiment Analysis in Hindi. *Webology, 19*(1), 592-600.
16. Navigli, R., & Crisafulli, G. (2010). Inducing word senses to improve web search result clustering. *Information Processing & Management, 46*(2), 213-227.
17. Padwad, H., Keswani, G., Bisen, W., Sharma, R., Thakre, S., & Tiwari, A. (2024). Leveraging Contextual Factors for Word Sense Disambiguation in Hindi Language. *International Journal of Intelligent Systems and Applications in Engineering, 12*(12s), 129-136.
18. Pal, A. R. (2017). A knowledge based methodology for word sense disambiguation for low resource language. *Advances in Computational Sciences and Technology,, 10*(2), 267-283.
19. Pal, A. R. (2017, February). Word sense disambiguation in Bengali: An unsupervised approach. *In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) , IEEE.*, (pp. 1-5). Coimbatore.
20. Pandit, R. S. (2018, January). Improving Lesk by Incorporating Priority for Word Sense Disambiguation. *In 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT). IEEE*, (pp. 1-4). Shibpur, India.
21. S. Mulkalapalli, B. P. (2016, December). Word Sense Disambiguation Techniques for Indian and other Asian Languages: A Survey. *International Journal of Computer Applications, 156*(8), 35-41.
22. S. S. Pandya, N. B. (2021). Preprocessing Phase of Text Sequence Generation for Gujarati Language. *Proceedings of the Fifth International Conference on Computing Methodologies and Communication*, (pp. 749-752). Erode, India. doi:doi: 10.1109/ICCMC51019.2021.9418046
23. Saha, P., Das Mou, A., & Mittra, T. (2019, 10). A Bangla Word Sense Disambiguation Technique using Minimum Edit Distance Algorithm and Cosine Distance. *23rd International Computer Science and Engineering Conference (ICSEC)* (pp. 1-6). IEEE.
24. Sarmah, J. &. (2016, May). Decision tree based supervised word sense disambiguation for Assamese. *Internationl Journal of Computer Applications, 141(1)., 141*(1), 42-48.
25. Sau, A., Amin, T., Barman, N., & Pal, A. (2019, 5). Word sense disambiguation in bengali using sense induction. *Proceedings - 2019 International Conference on Applied Machine Learning, ICAML 2019* (pp. 170-174). Institute of Electrical and Electronics Engineers Inc.
26. Soni, V., Gopalaniî, D., & Govil, M. (2021). An adaptive approach for word sense disambiguation for Hindi language. *IOP Conference Series: Materials*, *1131*.
27. Tayal, D. K. (2015). Word Sense Disambiguation in Hindi Language Using Hyperspace Analogue to Language and Fuzzy C-Means Clustering. *Proceedings of the 12th International Conference on Natural Language Processing* (pp. 49-58). Trivandrum, India: NLP Association of India.
28. Yusuf, M., Surana, P., & Sharma, C. (2022). HindiWSD: A Package for Word Sense Disambiguation in Hinglish & Hindi. *Proceedings of the WILDRE-6 Workshop @LREC2020*, (pp. 18-23). Marseille