SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I7P104

Volume 13 Issue 7, 51-61, July 2026

Original Article

Enhancing Hindi Word Sense Disambiguation Using Supervised Logistic Regression and Contextual Features

Vinto[1], Neeru Mago[2], Raj Kumari[3]

[1]*DCSA, Panjab University, Chandigarh, India.*
[2]*DCSA, PUSSGRC, Hoshiarpur, India.*
[3]*UIET, Panjab University, Chandigarh, India.*

[1]*Corresponding Author : vntgujjar@gmail.com*

Received: 18 March 2026 Revised: 21 April 2026 Accepted: 17 June 2026 Published: 29 July 2026

Abstract *- Natural languages are ambiguous by nature. Word sense ambiguity is one of the numerous layers of ambiguity. In many applications of natural language processing, sense ambiguity resolution is essential. In this work, word sense ambiguity is addressed, and a supervised method for Hindi word sense disambiguation has been suggested. A supervised method for Hindi word sense disambiguation is employed, incorporating contextual feature modeling and systematic preprocessing to effectively resolve ambiguity. After applying tokenization, POS tagging, stop-word removal, and Lemmatization, a context window is constructed over open-class words. A logistic regression model is used, and to extract feature TF-IDF technique is used. An average accuracy of 78.93% is shown by experimental findings on 20 polysemous Hindi words, which is higher than previous published work on the same dataset.*

Keywords *- Hindi Language, Word Sense Disambiguation, Supervised techniques, Knowledge-based, Unsupervised techniques.*

1. Introduction

Natural language text disseminated through the web is inherently ambiguous, as many words have several meanings that change depending on Context. While humans may easily infer the intended meaning based on contextual and language knowledge, automated systems encounter significant hurdles due to the lack of innate semantic comprehension.

Word Sense Disambiguation (WSD) seeks to overcome this issue by computationally determining the most appropriate sense of a word within its Context [1]. Despite decades of effort, WSD remains an open and tough problem in Natural Language Processing (NLP), and it is frequently referred to as an AI-complete task, implying that a general and totally dependable solution has yet to be found [2].

The complexity of WSD becomes more apparent in morphologically rich but resource-constrained languages like Hindi. The language has considerable inflection, unrestricted word order, and a high level of polysemy, making contextual interpretation difficult for automated models. Context is crucial for precise sense identification.

For example, there are two sentences in Hindi language-

1. हर समस्या का कोई न कोई हल ज़रूर होता है।
   (Every problem has a solution.)
2. किसान खेत में हल चला रहा था।
   (The farmer was ploughing the field.)

The common word " हल " in both phrases has varied meanings depending on the Context. It means "the solution/answer" in the first sentence and "a plough (farming tool)" in the second. Humans have no trouble figuring out a word's precise meaning. However, it is a challenging assignment for machines.

The WSD is especially applicable to real-world NLP applications where semantic precision is crucial. Improved Hindi WSD has a direct impact on practical systems such as machine translation [3], sentiment analysis [4], information retrieval [5], question answering [6], and grammatical analysis [7], where incorrect sense selection can result in mistranslations, irrelevant search results, or inaccurate sentiment classification. Despite significant research in NLP, progress in Hindi WSD remains relatively limited compared with resource-rich languages. Despite the fact that Hindi is one of the most commonly spoken languages in the world, the creation of reliable WSD systems for Hindi is still in the development stage. Numerous significant research gaps are shown by existing studies. First, the use of standardized benchmark datasets has received little attention, which makes fair comparison among different approaches difficult. Second, despite the fact that word lemmatization might enhance

semantic representation and lessen morphological changes, many current Hindi WSD research do not use it during preprocessing. Third, correct sense disambiguation is still severely hampered by Hindi's language complexity, which includes idiomatic expressions, morphological richness, and contextual ambiguity.

The novelty of the current work lies in addressing these identified research gaps for Hindi WSD using a methodical and supervised approach. To guarantee consistency, repeatability, and useful comparison with current methods documented in the literature, a standardized dataset has been used. Moreover, preprocessing has included word Lemmatization to enhance contextual feature representation and normalize inflected word forms. Additionally, to learn discriminative contextual patterns and enhance disambiguation performance, a supervised classification framework based on Logistic Regression has been utilized. The combination of supervised learning, language normalization via Lemmatization, and a common benchmark dataset improves classification accuracy and offers a more trustworthy framework for comparison evaluation in Hindi WSD. The experimental results show that the proposed methodology improves WSD performance, which contributes to the creation of more accurate and dependable Hindi language technology.

Sense1: **समाधान, निबटारा, निपटारा, हल, निराकरण, अपाकरण - सोच-समझकर ठीक निर्णय करने या परिणाम निकालने की क्रिया**

"मेरी समस्या का समाधान हो गया"

Sense 2: **हल, लाँगल, लांगल, सीर, कुंतल, कुन्तल, गाकील, नागल, नाँगल, सारंग - जमीन जोतने का एक उपकरण**

"किसान खेत में हल चला रहा है"

English Translation:

Sense 1: Solution, settlement, resolution, remedy, clarification — *the act of carefully thinking through a situation to arrive at a correct decision or outcome.*
"My problem has been resolved."
Sense 2: Plough — *an agricultural implement used for tilling or turning the soil.*
"The farmer is ploughing the field."

Fig. 1 Senses of "हल" (Hal)

Early research in Word Sense Disambiguation (WSD) was primarily knowledge-based, with Lesk's algorithm [8] being among the first and most significant approaches. The availability of large-scale sense inventories, lexical resources, and sense-annotated datasets, mainly in English, has greatly aided the development of such approaches. However, these resources are heavily language-dependent, require significant manual work to create, and frequently lack the semantic depth required for advanced concept-driven NLP applications [9]. To solve these problems, machine learning-based techniques have been proposed that use textual corpora to obtain the knowledge needed for disambiguation.

Among machine learning techniques, supervised WSD algorithms have performed well by learning explicit mappings between contextual variables and word senses. When there is a sufficient amount of sense-tagged training data, these approaches often perform well. However, creating large, manually annotated corpora is time-consuming and labor-intensive, making supervised WSD challenging to apply to low-resource languages like Hindi and other Indian languages with limited sense-annotated datasets. This data scarcity is a significant constraint for supervised learning-based WSD systems.

To address this issue, researchers investigated methods for augmenting current language resources and expanding existing knowledge bases. For example, the work published in [9] established KnowNet, a large-scale semantically enhanced knowledge base built using topic signatures collected from the web, which increased performance by combining WordNet, Extended WordNet, and KnowledgeNet. Ponzeto and Navigli [10] improved WordNet by automatically extracting semantic relations from Wikipedia, achieving results equivalent to cutting-edge supervised WSD systems on open-text evaluation tasks. Other researchers have used lexical information in learning-based frameworks to increase disambiguation accuracy [11, 12].

Given the availability of Hindi WordNet [13], supervised Word Sense Disambiguation (WSD) techniques that combine sense-annotated corpora with WordNet's lexical-semantic properties offer a potential option for Hindi. The study uses a target-word supervised WSD framework with the goal of improving disambiguation accuracy through thorough preprocessing and contextual feature optimization. A fixed context window is used to capture semantically relevant neighboring words, while essential preprocessing steps—like tokenization, stop word removal, Lemmatization, and Part-of-Speech (PoS) tagging—are used to reduce noise, normalize lexical variations, and improve feature discriminability.

A sense-annotated dataset with twenty polysemous phrases is used for assessment. An overall average accuracy of 20 words has been attained, surpassing the techniques reported in [14, 15]. The method in [14] involves computing the shortest-path similarity scores between word pairs, constructing a weighted graph G(V, E) where semantic relations are represented as weighted edges and synsets serve as vertices, and then utilizing a random walk algorithm on the graph to determine the most appropriate word senses, while [15] uses semantic similarity measures derived from Hindi WordNet and assesses the same dataset. However, they used a graph-based and similarity-based approach to disambiguate, in contrast to the current study. A publicly accessible corpus serves as the foundation for the evaluation dataset [16].

The experimental arrangement seeks to achieve robust disambiguation performance while eliminating the need for heavy manual annotation. The rest of the sections of the paper are arranged as follows: The following part discusses related literature, followed by a full discussion of the suggested technique, experimental evaluation, and concluding remarks.

2. Related Work

One of the more challenging tasks in NLP research is WSD. It is one of the earliest computational linguistics challenges. The late 1940s marked the start of this field's research. While Hindi WSD is still in its infancy, a large amount of work has been done for the English language. Sinha et al [17] presented the idea of disambiguating Indian languages. They devised a technique that involved comparing contexts containing unclear terms to those constructed using Hindi WordNet. The sense would be decided by the degree and breadth of overlap. Word meanings are used by the Hindi WordNet (HWN) to organize lexical data. The English WordNet had an impact on the design of the Hindi WordNet (HWN). HWN was created by IIT Bombay and made available to the general public in 2006. The accuracy ranges from 40% to 70%.

Several research has investigated Hindi WSD utilizing unsupervised, knowledge-based, supervised, and hybrid methods. Mishra et al. [18] proposed an unsupervised method that uses stop word removal and stemming during preprocessing, followed by the usage of decision lists for disambiguation, with just minimal manual involvement required to give seed examples. Another unsupervised approach is used in [15]. To distinguish between polysemous words, the authors utilized a semantic similarity score calculated using WordNet hierarchy. They obtained an overall average accuracy of 60.65% using a sense-annotated sample of 20 polysemous Hindi nouns.

Singh et al. [19] examined the effect of stemming, stop word removal, and context window size on a Lesk-like algorithm for Hindi WSD, finding that the combination of stop-word removal and stemming increased precision by 9.24% over the baseline, with Karak relations contributing to more accurate sense resolution. Graph-based WSD techniques have also been extensively studied. In [20], a graph-centric technique combining Lesk-based semantic similarity and indegree centrality was presented, with nodes representing word senses and edges encoding semantic relations extracted from Hindi WordNet, yielding an accuracy of 65.17%. Similarly, [15] used Leacock-Chodorow semantic relatedness to exploit hierarchical linkages in Hindi WordNet, yielding 60.65% accuracy on average using a dataset of 20 polysemous words. Singh et al. [21] investigated the impact of semantic relations such as hypernymy, hyponymy, holonymy, and meronymy, finding a 12.09% improvement in precision when all links were included, with hyponymy offering the greatest individual benefit.

Supervised learning algorithms have performed well when annotated data is available. In [22], a Naïve Bayes classifier including contextual, morphological, and syntactic features was assessed on 60 ambiguous Hindi nouns, attaining a highest precision of 86.11%. This highlights the importance of neighboring nouns in disambiguation. In [23], dynamic context window-based Lesk methods were investigated, and results showed that they outperformed fixed-window approaches. An unsupervised network-agglomeration-based WSD technique for Hindi is proposed in [24], which builds a sentence-level graph to represent all possible sense interpretations. Network agglomeration is used on the resulting interpretation graphs, and the interpretation with the largest agglomeration value is chosen as the best-suited meaning. In [25], an unsupervised graph-based framework is proposed that works in two stages: first, a graph is generated from a lexical knowledge base that captures all possible word senses and their semantic dependencies, and second, graph analysis techniques are used to identify the most adequate sense for each word within the given Context. In [14], the method entails calculating the shortest-path similarity scores between word pairs, building a weighted graph G(V, E) with semantic relations represented as weighted edges and synsets as vertices, and then using a random walk algorithm on the graph to identify the most suitable word senses.

Furthermore, Gautam et al. [26] used a Lesk-based technique with bigram and trigram contexts to disambiguate Hindi verbs, which is a very rare addition given that most extant Hindi WSD research focuses on noun disambiguation. Research has been expanded to include fuzzy and embedding-based models. Fuzzy Hindi WordNet (FHWN) was proposed to describe partial semantic relations using fuzzy membership values, and fuzzy graph connectivity metrics were used in [27], yielding in an 8% gain in performance. To lessen sensitivity to membership fluctuations, [28] investigated ConceptNet-based fuzzy centrality with Shapley values. ConceptNet, a multilingual knowledge base, is used to model words, senses, and phrases as interconnected nodes inside a semantic network in order to automatically compute membership values for fuzzy relationships. Additionally, the impact of differences in fuzzy associations was lessened by using the Shapley Value as a centrality metric to assess the marginal contribution of each membership value.

In [29], a supervised cosine similarity-based WSD utilizing Hindi WordNet was suggested, and it achieved 78.99% precision on 90 ambiguous words. Kumari and Lobiyal [30] suggested a word-embedding-based solution for WSD based on Word2Vec architectures, specifically continuous bag-of-words and skip-gram, in which cosine similarity is used to choose the most appropriate sense. In [31], Goonjan et al. extracted probable senses from Hindi WordNet and built a sense graph using depth-first search, followed by edge weighting based on Fuzzy Hindi WordNet relations; local fuzzy centrality metrics were then used to find

the correct sense. A knowledge-based Lesk algorithm based on Hindi WordNet was published in [32], obtaining a 71.43% accuracy by correctly disambiguating 2143 out of 3000 ambiguous sentences.

Tripathi et al. [33] improved the classic Lesk algorithm by providing a new scoring mechanism that analyzes coherent variants of token senses based on gloss information and associated hypernym and hyponym relationships. A further study by Kumari and Lobiyal [34] used Hindi Wikipedia data to compare different embedding algorithms, including BoW, TF-IDF, Word2Vec, and FastText, and found that Word2Vec consistently outperformed the others. Their framework also incorporates clustering to create a sense inventory, allowing for unsupervised disambiguation of unlabeled data. Several frameworks and hybrid techniques have also been proposed. The "hindiwsd" framework uses a modified Lesk algorithm to provide a complete pipeline for transliteration, spell correction, POS tagging, and WSD, with a 71% accuracy. A Hindi WSD method based on a Genetic Algorithm was presented in [36]. Selection, crossover, and mutation operations for sense disambiguation came after preprocessing, context bag production, and sense bag generation. The approach obtained an accuracy of 80% when tested on a manually created dataset.

An unsupervised graph-based method for Hindi WSD was presented in [37]. A random walk methodology was used for sense disambiguation after a graph representing the many senses of polysemous words was created. Path-based metrics were used to identify semantic similarity across nodes. The Leacock–Chodorow Similarity fared better than the Shortest Path measure, with an average accuracy of 72.09% across five polysemous terms, according to experimental data. A weighted graph-based method for Hindi WSD was reported in [14], in which semantic associations obtained from Hindi WordNet were modeled as weighted edges, and word senses were represented as nodes. Evaluation on a sense-annotated dataset of 20 polysemous words yielded an overall accuracy of 63.39%, surpassing previously reported approaches on the same dataset, using a random-walk algorithm to choose the most appropriate meaning in Context.

An unsupervised graph-based method for Hindi WSD was given in [38], where semantic relations from Hindi WordNet were used to represent synsets of the target and context words as nodes in a weighted directed graph. The Leacock–Chodorow Similarity was used to calculate edge weights, and graph centrality techniques were used for sense selection. On 20 ambiguous Hindi nouns, PageRank achieved the greatest accuracy of 66.92% from the four different types of graph-based centrality algorithms. In order to solve the lack of Hindi language resources and capture contextual semantic information, a deep learning-based method for Hindi WSD was presented in [39] by combining Word2Vec and FastText with a Bi-directional LSTM network. The model received an F1-score of 70.03% when tested on a sense-annotated corpus of 70 extremely ambiguous Hindi words.

3. Proposed Methodology

The logistic regression-based model for disambiguation of word sense for target words is described in this section, along with the method used to identify the most appropriate sense for a target word in a particular sentence. Logistic Regression is used as the main classification model in a target-word supervised Word Sense Disambiguation (WSD) framework for the Hindi language. Logistic Regression was chosen because of its stability under limited annotated data—a major limitation in Hindi WSD research—strong generalization ability in high-dimensional sparse feature spaces, and interpretability of learnt weights. The pipeline for research methodology is shown in Figure 2. The suggested methodology's algorithmic steps are:

3.1. Dataset

The Sense-Annotated Corpus for Hindi [16] is a curated linguistic resource specifically developed for lexical sample–based WSD in Hindi. It consists of 60 ambiguous Hindi nouns and has been designed to facilitate the systematic evaluation of WSD methodologies for Indian languages. The corpus is made available via the TDIL (Technology Development for Indian Languages) portal, a government-supported initiative under the Ministry of Electronics and Information Technology, Government of India. By offering manually annotated sense information, the dataset serves as a foundational benchmark for advancing research in Hindi and other Indian language processing tasks. Twenty ambiguous nouns from the dataset have been used in the experiment.

3.2. Problem Formulation

Let w_t stand for a target polysemous Hindi word that appears in sentence S. WSD's objective is to allocate the most suitable sense s_i to w_t based on its surrounding Context, given a collection of predefined senses $\{s_1, s_2 \ldots s_k\}$ from Hindi WordNet. Each training instance is represented as:

$$(S, w_t, s_i)$$

where s_i is the standard sense label derived from a sense-annotated Hindi corpus.

Each sense of the target word is associated with a different class in the given task, which is described as a multiclass classification issue.

3.3. Preprocessing

Each sentence is subjected to the following preprocessing procedures in order to minimize linguistic noise and normalize lexical variations:

- Tokenization: Indic NLP tokenizers are used to divide sentences into word tokens.

- Stop-word Removal: In order to preserve tokens that are semantically informative, function words (such as "का", "के", "है") are eliminated.
- Lemmatization: To deal with the rich morphology of Hindi, inflected word forms are reduced to their base forms.
- Part-of-Speech (PoS) Tagging: To assist in sense detection, each token is annotated with its grammatical category. These steps ensure that the extracted features emphasize meaningful contextual cues relevant to sense disambiguation.

3.4. Creation of Context Window

A fixed context window with a size of ±5 is used. It indicates that a maximum of five words are taken into account on the target word's left and right sides:

$$C= \{w_{-n}, w_{-n+1}..., w_{-2}, w_{-1}, w, w_1, w_2..., w_{n-1}, w_n\} \quad (1)$$

The window preserves computational efficiency while capturing adequate semantic Context.

3.5. Feature Extraction

An advanced and popular text representation method in NLP and information retrieval is called TF-IDF (Term Frequency-Inverse Document Frequency). The TF component establishes a term's local relevance by counting how often it occurs in a given document. The IDF component, on the other hand, evaluates a term's uniqueness or usefulness across the corpus. Words with multiple occurrences have lower IDF scores since they are less useful in differentiating between texts. Multiplying the TF and IDF values yields the final TF-IDF score, which is a weighted representation that suppresses common words while highlighting rare but significant ones.

The following is a description of TF (Term Frequency), which quantifies how often a phrase appears in a document.

$$TF(t,d) = \frac{f(t,d)}{|d|} \quad (2)$$

Where f(t,d) is the frequency of term t in document d, and |d| is the total number of terms in the document.

IDF (Inverse Document Frequency), which is defined as follows, lessens the weight of terms that appear often across documents:

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \quad (3)$$

Where df(t) is the number of documents that include the term t, and N is the total number of documents in the corpus.

The final TF-IDF weight is calculated as follows:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \quad (4)$$

Due to its effectiveness and simplicity, TF–IDF is extensively used in text classification, document clustering, information retrieval, and sentiment analysis tasks; that's why the TF-IDF technique is used here for feature extraction.

3.6. Classification Using Logistic Regression

As a supervised multiclass classifier, logistic Regression is used. Given a feature vector x, the softmax function determines the probability of assigning sense s_k.

$$P = (y = s_k|x) = \frac{e^{w_k^t x}}{\sum_{j=1}^{k} e^{w_j^t x}} \quad (5)$$

Where K is the target word's total number of senses, x is the feature vector that represents the target word's Context, w_k is the weight vector related to sense s_k, $w_k^t x$ is the score that indicates how well the Context supports sense s_k, and $P(y=s_k|x)$ is the likelihood that the target word takes sense s_k.

The maximum probability rule is used to choose the predicted sense after the probability distribution across all senses has been determined:

$$\hat{s} = argmax_k P(y = s_k|x) \quad (6)$$

3.7. Model Training and Evaluation

From the dataset, training and testing sets have been created, and performance is evaluated using accuracy. A commonly used technique in supervised machine learning and natural language processing research, the dataset has been divided into 80% for training and 20% for testing in order to assess the suggested Hindi Word Sense Disambiguation (WSD) model.

The division ensures that enough data are available for training while keeping a reasonable percentage of unseen instances for performance evaluation, offering a fair trade-off between model learning and objective evaluation.

4. Performance Evaluation

4.1. Dataset

Twenty ambiguous nouns from a publicly accessible sense-annotated dataset have been used in the experiment. Twenty ambiguous terms from the baseline are used to evaluate the suggested approach. In total, 965 instances have been evaluated across all target words. There are roughly 48.25 times per word and 20.10 instances each sense on average.

The dataset allows to compare the proposed work to earlier studies that were evaluated using the same dataset without requiring additional effort [14, 15]. For discrimination, the baseline studies use a graph-based method [14], and an unsupervised similarity-based method [15].

4.2. Experimental Setup

A supervised Logistic Regression classifier in combination with TF-IDF-based contextual feature extraction was implemented to develop the suggested Hindi Word Sense Disambiguation model. To capture the contextual significance of terms around the target word, the contextual instances were converted into numerical feature vectors using term frequency–inverse document frequency representation. The regularization parameter C=0.1 was used to configure the classifier; a lower value of C applies stronger regularization and helps prevent overfitting by limiting unduly complex decision boundaries. The maximum number of optimization steps permitted for the learning algorithm to converge toward an optimal solution is defined by the maximum iteration limit, which was set at 500.

In order to minimize bias toward classes with more training examples, balanced class weighting was used to automatically give minority sense classes more weight. An 80:20 ratio with a constant random state of 42 was used to split the dataset into training and testing subsets, guaranteeing consistent data partitioning and repeatability of experimental results over several runs. Accuracy, precision, recall, and F1-score were among the common metrics used to assess model performance. Python 3.12.13 was used for all experiments in the Google Colab environment. For data preprocessing, feature extraction, model training, and evaluation, the implementation made use of popular libraries such as Scikit-learn, NumPy, and Pandas. In order to analyze the Hindi corpus efficiently and conduct reproducible experiments, the studies were carried out on a cloud-based computing platform with about 12 GB RAM and GPU capabilities.

4.3. Proposed Algorithm

The proposed Hindi WSD algorithm, as in Algorithm 1, employs a structured pipeline to transform unprocessed Hindi text into accurate sense predictions through a sequence of preprocessing, feature extraction, and classification stages. The input sentence containing an ambiguous Hindi term is first subjected to preprocessing methods such as tokenization, Lemmatization, stop-word removal, and PoS tagging in order to normalize linguistic variances. Contextual characteristics are subsequently extracted using the TF-IDF representation, which captures the significance of surrounding words in the given Context. After receiving these inputs, a Logistic Regression classifier forecasts the target word's most appropriate sense using learned decision bounds. The outcome reflects the disambiguated meaning of the ambiguous Hindi word.

Algorithm 1: Supervised Learning-Based Algorithm for Efficient Sense Prediction

Input: Set of target words TW={tw_1,tw_2,…,tw_n}, Corpus
Output: Predicted sense labels and performance metrics (Accuracy, Precision, Recall, F1-score)

Step 1: Initialize performance metrics list
Step 2: ***for*** each target word $tw \in TW$ do
Step 3: $df \leftarrow$Load_Files_From_Dir(tw)
Step 4: $X \leftarrow df$ [Context] + df [Target_Word]
Step 5: $y \leftarrow df$ [Sense]
Step 6: $X \leftarrow$Preprocess(X)
// includes stopword removal, tokenization, Lemmatization, etc.
Step 7: (X_{train}, X_{test} , y_{train} ,y_{test})$\leftarrow$Train_Test_Split(X,y)
Step 8: ***Initialize pipeline:***
Model $\leftarrow$ TF–IDF + Logistic Regression
Step 9: Model . ***fit*** (X_{train},y_{train})
Step 10: $y_{pred} \leftarrow$ Model. ***predict***(X_{test})
Step 11: Acc$\leftarrow$Accuracy(y_{test},y_{pred})
Precision$\leftarrow$Precision(y_{test},y_{pred})
Recall$\leftarrow$Recall(y_{test},y_{pred})
F1$\leftarrow$F1_Score(y_{test},y_{pred})
Step 12: ***store*** Results (tw, Acc, Precision, Recall, F1)
Step 13: ***end for***
Step 14: ***return***

A line-by-line explanation of Algorithm 1 is discussed below:

Step 1: Initialize a structure to store performance metrics for all target words.
Step 2: Iterate over each ambiguous target word in the dataset.
Step 3: Load the dataset corresponding to the current target word into a structured format.
Step 4: Construct input features by combining context sentences with the target word.
Step 5: Extract the corresponding sense labels as the target variable.
Step 6: Apply preprocessing techniques to clean and normalize the textual data.
Step 7: Split the dataset into training and testing subsets for evaluation.
Step 8: Initialize a pipeline combining a feature extraction technique with a classifier.
Step 9: Train the model using the training data.
Step 10: Predict sense labels for the test data using the trained model.
Step 11: Compute evaluation metrics, including Accuracy, Precision, Recall, and F1-score.
Step 12: Store the performance results for the current target word.
Step 13: Repeat the Process for all target words.
Step 14: Return the aggregated performance results of the model.

4.4. Evaluations and Results

First, the dataset is preprocessed. Tokenization and POS tagging have been done using a POS tagger built for Indian languages. The baseline is the study published in [14, 15]. A fixed context window size of ± 5 is used for the test run. First of all, tokenization is performed using the indicnlp and

indic_nlp_library frameworks, which offer specialized tools for Indian languages, allowing for tokenization, text normalization, and transliteration. The indic_tokenize module effectively segments text into major components like words and sentences. After that, stop words are removed using the CLTK library (Classical Language Toolkit), which is intended for use with historical and classical languages. It uses CorpusImporter to import corpora for Hindi language studies, with a concentration on Classical Hindi and early forms of Hindi writing. Stop word elimination, or removal, reduces noise in NLP jobs while increasing the efficiency of text analysis tools. Before extracting the context window, words are converted to their base form using Lemmatization. The Stanford NLP Group's Stanza, a Python-based natural language processing framework, is used for Lemmatization. After performing Lemmatization, a context window is created. When constructing Context, all open class words—nouns, adjectives, verbs, and adverbs—are taken into consideration.

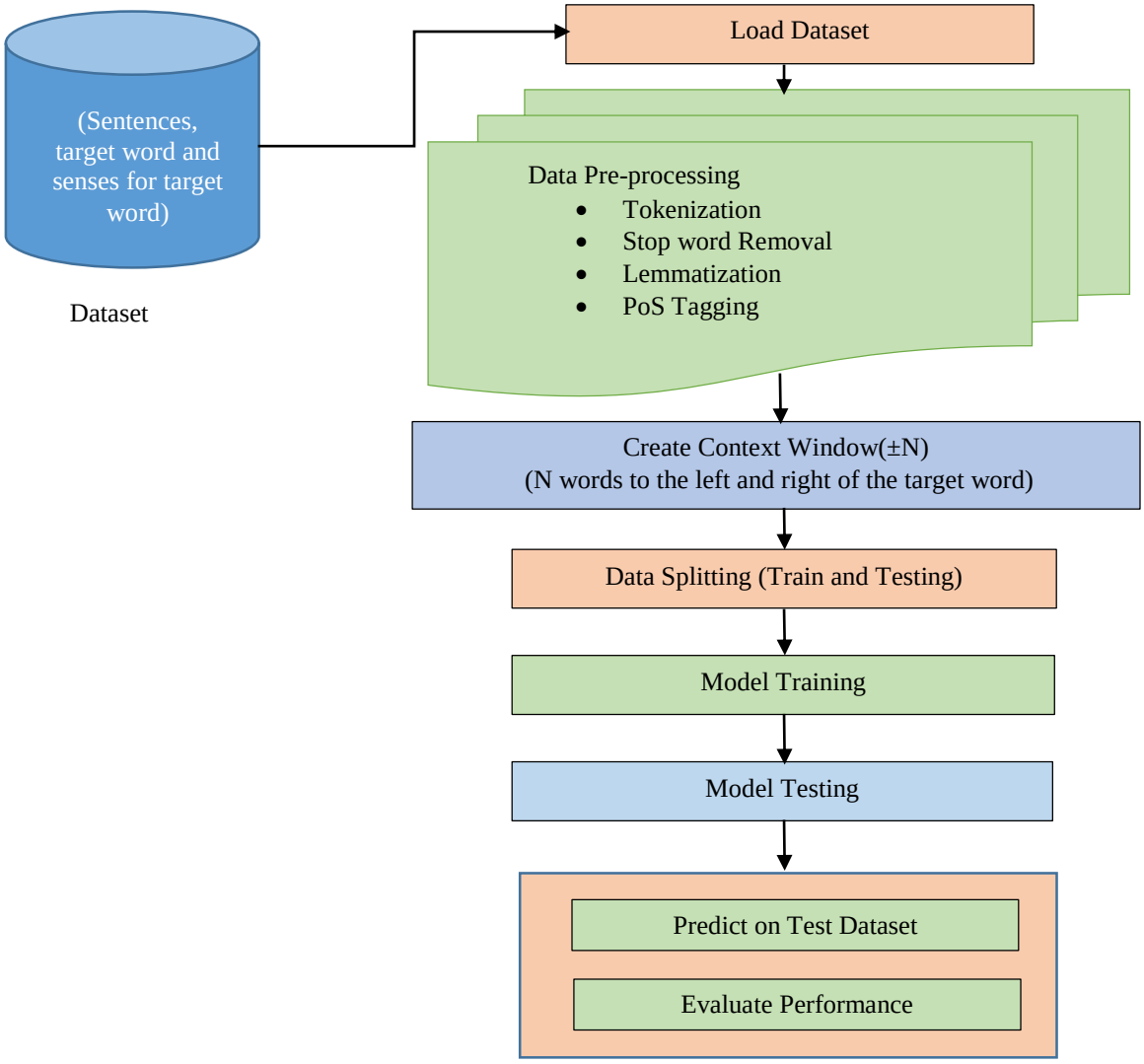

Fig. 2 Pipeline for research methodology

For each context window, features are extracted using the TF-IDF feature extraction technique. The logistic regression classifier is trained and tested using 80:20 ratios. For evaluation purposes, the accuracy is computed for each word. For each of its senses, the average accuracy of all instances is calculated. Table 1 displays the total number of instances, senses, and observed accuracy for each of the 20 polysemous terms.

Additionally, the correctness of each word is compared in the table with the accuracy stated in [14, 15].

Table 2 shows that the overall average accuracy achieved using the suggested approach is 78.93%, which is much better than what was reported in [14, 15].

Table 1. Accuracy comparison table for 20 words with methods working on same dataset

Word	No. of Senses	Total Instances	Accuracy of Proposed Algorithm	Accuracy[14]	Accuracy[15]
उत्तर	3	66 (#Is1-15, #Is2-35, #Is3-16)	0.7333	0.6060	0.9030
कुंभ	3	71 (#Is1-22, #Is2-22, #Is3-27)	**0.7778**	0.6338	0.5701
कोटा	2	58 (#Is1-33, #Is2-25)	0.6667	0.6724	0.6416
गुरु	2	44 (#Is1-26, #Is2-18)	**0.7778**	0.7272	0.6782
चंदा	2	54 (#Is1-20, #Is2-34)	**1.0**	0.7407	0.5687
जेठ	2	20 (#Is1-10, #Is2-10)	**1.0**	0.7	0.6666
डाक	2	36 (#Is1-14, #Is2-22)	**0.7778**	0.6944	0.5000
तान	2	29 (#Is1-13, #Is2-16)	0.3333	0.5862	0.4166
तीर	2	36 (#Is1-25, #Is2-11)	**1.0**	0.6388	0.7500
दाम	2	39 (#Is1-24, #Is2-15)	**0.875**	0.5897	0.7368
फल	3	50 (#Is1-23, #Is2-19, #Is3-8)	**0.9**	0.66	0.6526
माँग	2	46 (#Is1-24, #Is2-22)	**0.8333**	0.6087	0.6363
मूल	4	90 (#Is1-5, #Is2-35, #Is3-28, Is4-22)	**0.9444**	0.7	0.9019
वचन	3	28 (#Is1-9, #Is2-11, #Is3-8)	0.6667	0.6785	0.3809
विधि	2	83 (#Is1-53, #Is2-30)	0.5625	0.6265	0.5909
संक्रमण	3	56 (#Is1-14, #Is2-23, #Is3-19)	**0.7143**	0.5714	0.1809
संबंध	3	34 (#Is1-14, #Is2-14, #Is3-6)	**0.8571**	0.6176	0.3888
सोना	2	44 (#Is1-26, #Is2-18)	**0.875**	0.5909	0.7500
हल	2	43 (#Is1-15, #Is2-28)	**0.7778**	0.6744	0.6114
हार	2	43 (#Is1-15, #Is2-28)	**0.7143**	0.6758	0.6052
Average Accuracy			**0.7893**	0.6339	0.6065

Table 2. Accuracy comparison table with state-of-the-art techniques

WSD Type	Approach	Accuracy (%)
Knowledge Based	Lesk (Overlap Based) [17]	31.7
Un-Supervised	Fuzzy C-Means Clustering [40]	40-70
	Graph Based[14] **(Same Dataset)**	63.39
Neural Network Based	Word2Vec + Cosine Similarity[30]	52.00
	FastText + Cosine Similarity[34]	54.00
	Word2Vec+ Bi-LSTM[39]	63.29
	FastText + Bi-LSTM[39]	65.71
Supervised	Naïve Based [15]**(Same Dataset)**	60.65
	TF-IDF + Logistic Regression (Proposed Model)	**78.93**

4.5. Discussion

In comparison to the baseline procedures [14, 15], which yielded 63.39% and 60.65%, respectively, the proposed approach obtains an average accuracy of 78.93% over all instances of the 20 polysemous Hindi terms, as Table 1 illustrates. The findings show that for the majority of target terms, the suggested approach consistently performs better than current methods. The suggested method benefits from an enriched preprocessing strategy and contextual feature optimization, allowing for more successful word sense classification than the baseline methods, which rely on limited contextual information. The consistency of performance across various terms is a key observation from the table. The lowest accuracy recorded is 0.3333 for the word तान, while the highest accuracy of 1.0 is attained for a number of words, including चंदा, जेठ, and तीर. The baseline approaches, on the other hand, exhibit more fluctuation and far lower minimum accuracies for a number of terms, suggesting less consistent behavior. The consistency implies that the suggested approach is less susceptible to different numbers of senses and skewed sense distributions. The proposed approach excels in

disambiguating highly polysemous words; for instance, it achieved an accuracy of 94.44% for मूल (four senses) and 73.33% for उत्तर (three senses). The performance boost stems from optimized preprocessing and contextual representations that strengthen the model's discriminative capabilities. While performance dips slightly for terms lacking rich contextual cues, the overall results validate the method's robustness. The proposed algorithm outperforms the approaches in [14, 15] by providing superior flexibility across different lexical types, whereas previous models remain constrained by fixed contextual measures. It demonstrates that the proposed method is a robust, resource-efficient solution for Hindi WSD, proving particularly effective when data is scarce. Comparison of the accuracy of the proposed model with state-of-the-art techniques is given in Table 2. This data illustrates that the proposed model is superior to other methodologies that have been used for HWSD on the same dataset. Several major elements contribute to the better performance of the proposed supervised strategy over graph-based and Naïve Bayes-based algorithms. One of the key reasons for the higher classification accuracy is the use of Lemmatization during the preprocessing stage, which was not explored in the competing approaches. Hindi is a morphologically rich language in which a single root word can have several inflected forms depending on gender, number, tense, and grammatical Context. Using Lemmatization, distinct morphological forms are normalized into their root form, minimizing vocabulary sparsity and enhancing semantic consistency within the Context.

This allows the classifier to learn more meaningful contextual patterns for each sense of the ambiguous word. In contrast, graph-based techniques rely heavily on semantic relationships and graph traversal processes derived from lexical resources, whereas Naïve Bayes assumes conditional independence among contextual features, which may restrict its capacity to capture complicated contextual dependencies. The combination of language normalization via Lemmatization, contextual feature extraction, and supervised learning thus greatly adds to the proposed method's higher performance over previous techniques.

5. Conclusion and Future Scope

Supervised learning techniques have become the most successful paradigm for attaining high disambiguation accuracy in the field of Hindi WSD. The three main types of current WSD techniques are knowledge-based, unsupervised, and supervised. A clear trade-off between both paradigms is revealed by comparative studies: supervised approaches routinely outperform others because of their excellent discriminative capabilities, but their efficacy is strongly correlated with the availability of sense-annotated corpora. Unsupervised methods, on the other hand, do away with the need for labelled data, but they frequently have poor precision and contextual sensitivity. In order to better comprehend the work of diverse researchers, a comparative examination of alternative methodologies has also been conducted.

To ensure robustness and reproducibility, the suggested supervised logistic regression-based method can be further developed by testing it on bigger, standardized Hindi WSD datasets. Future research might concentrate on adding better contextual and lexical elements to the feature set while keeping logistic Regression's ease of use and interpretability. The method can also be modified to serve as a robust baseline model for evaluating Hindi WSD on common datasets and for comparison with more sophisticated supervised and hybrid approaches in practical NLP applications. Future work may also focus on extending the Hindi word sense disambiguation dataset with more ambiguous words and contextual instances from other common linguistic resources and real-world data. The updated dataset could also be maintained in a balanced or structured way to make it easier to access, replicate, and compare across different evaluations of the same dataset. Additionally, advanced deep learning models such as CNN, Bi-LSTM, and BERT could be used to the expanded dataset to enhance their performance in disambiguating Hindi words.

Declarations

Conflict of Interest

The corresponding author declares that there is no conflict of interest on behalf of all authors.

References

[1] Roberto Navigli, "Word Sense Disambiguation: A Survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1-16, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yorick Wilks, and Dann Fass, "The Preference Semantics Family," *Computers & Mathematics with Applications*, vol. 23, no. 2-5, pp. 205-221, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[3] M. Carpuat, and D. Wu, "Improving Statistical Machine Translation using Word Sense Disambiguation," *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic, pp. 61-72, 2007. [Google Scholar] [Publisher Link]

[4] Chiraag Sumanth, and Diana Inkpen, "How Much Does Word Sense Disambiguation Help in Sentiment Analysis of Micropost Data?," *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Lisboa, Portugal, pp. 115-121, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] C. Carpineto, and Giovanni Romano, "A Survey of Automatic Query Expansion in Information Retrieval," *ACM Computing Surveys*, vol. 44, no. 1, pp. 1-50, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[6] Adrian-Gabriel Chifu, and Radu-Tudor Ionescu, "Word Sense Disambiguation to Improve Precision for Ambiguous Queries," *Central European Journal of Computer Science*, vol. 2, pp. 398-411, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[7] V. Advaith, Anushka Shivkumar, and B.S Sowmya Lakshmi, "Parts of Speech Tagging for Kannada and Hindi Languages using ML and DL Models," *2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, pp. 1-5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Michael Lesk, "Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone," *Proceedings of the 5th Annual International Conference on Systems Documentation*, pp. 24-26, 1986. [CrossRef] [Google Scholar] [Publisher Link]

[9] Montse Cuadros, and German Rigau, "KnowNet: Building a Large Net of Knowledge from the Web," *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 161-168. 2008. [Google Scholar] [Publisher Link]

[10] Simone Paolo Ponzetto, and Roberto Navigli, "Knowledge-Rich Word Sense Disambiguation Rivaling Supervised Systems," *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 1522-1531, 2010. [Google Scholar] [Publisher Link]

[11] Eneko Agirre, and Aitor Soroa, "SemEval-2007 Task 02: Evaluating Word Sense Induction and Discrimination Systems," *Proceedings of the Fourth International Workshop on Semantic Evaluations*, Prague, Czech Republic, pp. 7-12, 2007. [Google Scholar] [Publisher Link]

[12] Ravi Sinha, and Rada Mihalcea, "Unsupervised Graph-basedWord Sense Disambiguation Using Measures of Word Semantic Similarity," *International Conference on Semantic Computing (ICSC 2007)*, Irvine, CA, USA, pp. 363-369, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[13] Pushpak Bhattacharyya, *IndoWordnet*, The WordNet in Indian Languages, pp. 1-18, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[14] Prajna Jha et al., "A Novel Unsupervısed Graph-Based Algorıthm for Hindi Word Sense Disambiguation," *SN Computer Science*, vol. 4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Satyendr Singh, Vivek Kumar Singh, and Tanveer J. Siddiqui, "Hindi Word Sense Disambiguation Using Semantic Relatedness Measure," *7th International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, Berlin, Heidelberg, pp. 247-256, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[16] Satyendr Singh, and Tanveer J. Siddiqui, "Sense Annotated Hindi Corpus," *2016 International Conference on Asian Language Processing (IALP)*, Tainan, Taiwan, pp. 22-25, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[17] Manish Sinha et al., Hindi Word Sense Disambiguation, Department of Computer Science and Engineering, pp. 1-7, 2004. [Online]. Available: https://www.cse.iitb.ac.in/~pb/papers/HindiWSD.pdf

[18] Neetu Mishra, Shashi Yadav, and Tanveer J. Siddiqui, "An Unsupervised Approach to Hindi Word Sense Disambiguation," *Proceedings of the First International Conference on Intelligent Human Computer Interaction*, pp. 327-335, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[19] Satyendr Singh, and Tanveer J. Siddiqui, "Evaluating Effect of Context Window Size, Stemming and Stop Word Removal on Hindi Word Sense Disambiguation," *2012 International Conference on Information Retrieval & Knowledge Management*, Kuala Lumpur, Malaysia, pp. 1-5, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[20] Sandeep Kumar Vishwakarma, and Chanchal Kumar Vishwakarma, "A Graph Based Approach to Word Sense Disambiguation for Hindi Language," *International Journal of Scientific Research Engineering & Technology*, vol. 1, no. 5, pp. 313-318, 2012. [Google Scholar] [Publisher Link]

[21] Satyendr Singh, and Tanveer J. Siddiqui, "Role of Semantic Relations in Hindi Word Sense Disambiguation," *Procedia Computer Science*, vol. 46, pp. 240-248, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[22] Satyendr Singh, Tanveer J. Siddiqui, and Sunil K. Sharma, "Naïve Bayes Classifier for Hindi Word Sense Disambiguation," *Proceedings of the 7th ACM India Computing Conference*, pp. 1-8, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[23] Radhike Sawhney, and Arvinder Kaur, "A Modified Technique for Word Sense Disambiguation using Lesk Algorithm in Hindi language," *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Delhi, India, pp. 2745-2749, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[24] Amita Jain, and D.K Lobiyal, "Unsupervised Hindi Word Sense Disambiguation based on Network Agglomeration," *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 195-200, 2015. [Google Scholar] [Publisher Link]

[25] Lokesh Nandanwar, "Graph Connectivity for Unsupervised Word Sense Disambiguation for Hindi Language," *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, pp. 1-4, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[26] Chandra Bhal Gautam, and Dilip Kumar Sharma, "Hindi Word Sense Disambiguation using Lesk Approach on Bigram and Trigram Words," *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, pp. 1-5, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[27] Amita K. Jain, and Daya Krishan Lobiyal, "Fuzzy Hindi WordNet and Word Sense Disambiguation Using Fuzzy Graph Connectivity Measures," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 15, no. 2, pp. 1-31, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[28] Goonjan Jain, and Daya Krishan Lobiyal, "Word Sense Disambiguation using Cooperative Game Theory and Fuzzy Hindi WordNet based on ConceptNet," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 4, pp. 1-25, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Sarika, and Dilip Kumar Sharma, "Hindi Word Sense Disambiguation Using Cosine Similarity," *Proceedings of International Conference on ICT for Sustainable Development*, pp. 801-808. 2016. [CrossRef] [Google Scholar] [Publisher Link]

[30] Archana Kumari, and D.K. Lobiyal, "Word2vec's Distributed Word Representation for Hindi Word Sense Disambiguation," *16th International Conference Distributed Computing and Internet Technology*, Bhubaneswar, India, pp. 325-335, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[31] Goonjan Jain, and D. K. Lobiyal, "Word Sense Disambiguation of Hindi Text Using Fuzzified Semantic Relations and Fuzzy Hindi WordNet," *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, pp. 494-497, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[32] Pooja Sharma, and Nisheeth Joshi, "Knowledge-Based Method for Word Sense Disambiguation by Using Hindi WordNet," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3985-3989, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[33] Praffullit Tripathi et al., "Word Sense Disambiguation in Hindi Language Using Score Based Modified Lesk Algorithm," *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 939-954, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[34] Archana Kumari, and D.K. Lobiyal, "Efficient Estimation of Hindi WSD with Distributed Word Representation in Vector Space," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6092-6103, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[35] Mirza Yusuf et al., "HindiWSD: A Package for Word Sense Disambiguation in Hinglish & Hindi," *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, Marseille, France, pp. 18-23, 2022. [Google Scholar] [Publisher Link]

[36] Surbhi Bhatia, Ankit Kumar, and Mohammed Mutillah Khan, "Role of Genetic Algorithm in Optimization of Hindi Word Sense Disambiguation," *IEEE Access*, vol. 10, pp. 75693-75707, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[37] Prajna Jha et al., "Comparative Analysis of Path-based Similarity Measures for Word Sense Disambiguation," *2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP)*, Vijayawada, India, pp. 1-5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Prajna Jha et al., "Unsupervised Hindi Word Sense Disambiguation using Graph- Based Centrality Measures," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 4, pp. 4957-4964, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[39] Binod Kumar Mishra, and Suresh Jain, "Word Sense Disambiguation for Indic Language using Bi-LSTM," *Multimedia Tools and Applications*, vol. 84, pp. 16631-16656, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[40] Devendra K. Tayal, Leena Ahuja, and Shreya Chhabra, "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*, Trivandrum, India, pp. 49-58, 2015. [Google Scholar] [Publisher Link]