information

Review

A Literature Survey on Word Sense Disambiguation for the Hindi Language

Vinto Gujjar [1], Neeru Mago [2], Raj Kumari [3], Shrikant Patel [4], Nalini Chintalapudi [5,*] and Gopi Battineni [5,6]

1 Department of Computer Science & Applications, Panjab University, Chandigarh 160014, India; vntgujjar@gmail.com
2 Department of Computer Science & Applications, Panjab University Swami Sarvanand Giri Regional Centre, Hoshiarpur 160014, India
3 University Institute of Engineering and Technology, Panjab University, Chandigarh 160014, India
4 School of IT & ITES, Delhi Skill and Entrepreneurship University, Government of NCT of Delhi, Delhi 110003, India
5 Clinical Research Centre, School of Medicinal and Health Products Sciences, University of Camerino, 62032 Camerino, Italy
6 Department of Electronics and Communication Engineering, Velagapudi Ramakrishna Siddharth Engineering College, Vijayawada 520007, India
* Correspondence: nalini.chintalapudi@unicam.it

Abstract: Word sense disambiguation (WSD) is a process used to determine the most appropriate meaning of a word in a given contextual framework, particularly when the word is ambiguous. While WSD has been extensively studied for English, it remains a challenging problem for resource-scarce languages such as Hindi. Therefore, it is crucial to address ambiguity in Hindi to effectively and efficiently utilize it on the web for various applications such as machine translation, information retrieval, etc. The rich linguistic structure of Hindi, characterized by complex morphological variations and syntactic nuances, presents unique challenges in accurately determining the intended sense of a word within a given context. This review paper presents an overview of different approaches employed to resolve the ambiguity of Hindi words, including supervised, unsupervised, and knowledge-based methods. Additionally, the paper discusses applications, identifies open problems, presents conclusions, and suggests future research directions.

Keywords: word sense disambiguation; knowledge-based; supervised; unsupervised; Hindi language

Citation: Gujjar, V.; Mago, N.; Kumari, R.; Patel, S.; Chintalapudi, N.; Battineni, G. A Literature Survey on Word Sense Disambiguation for the Hindi Language. *Information* **2023**, *14*, 495. https://doi.org/10.3390/info14090495

Academic Editor: Peter Revesz

Received: 7 July 2023
Revised: 30 August 2023
Accepted: 2 September 2023
Published: 7 September 2023

1. Introduction

In the present age of information technology (IT), the whole world is sharing information using the internet. This information is available in natural language. As naturally understood, all-natural languages have an intrinsic feature called ambiguity. Ambiguity refers to the situation where a word can have multiple meanings. Ambiguity in natural language poses a significant obstacle in Natural Language Processing (NLP). While the human mind can rely on cognition and world knowledge to disambiguate word senses, machines lack the ability to employ cognition and world knowledge, leading to semantic errors and erroneous interpretations in their output. Therefore, the WSD process is employed to alleviate ambiguity in sentences.

WSD represents highly regarded formidable challenges within the realm of NLP and stands as one of the earliest quandaries in computational linguistics. Experimentation efforts in this domain commenced in the late 1940s, with Zipf's [1] introduction of the "law of meaning" in 1949. This principle posits a power law relationship between the frequency of a word and the number of meanings it possesses, indicating that more common words tend to have a greater range of meanings compared to less frequent ones. In 1975, Wilks [2] advanced the field by developing a model known as "preference semantics", which employed selectional constraints and frame-based lexical semantics to ascertain the

precise meaning of a polysemous word. Notably, the 1980s witnessed significant progress in WSD research, facilitated by the availability of extensive lexical resources and corpora. Ultimately, WSD entails the task of identifying the accurate sense of a word within its specific contextual framework [3]. WSD is not considered a final objective; instead, it is recognized as an intermediary task with relevance to various applications within the field of NLP. Figure 1 presents the WSD conceptual diagram.

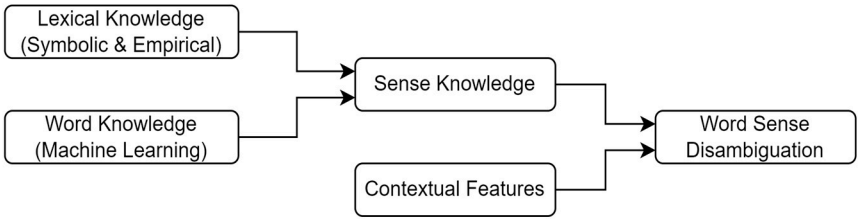

Figure 1. Conceptual Diagram of WSD.

In machine translation, WSD is an important step because a number of words in every language have a different translation according to the context of their usage [3–6]. It is an important issue to be considered during language translation. WSD assumes a crucial role in ensuring precise text analysis across a wide range of applications [7,8]. For example, an intelligence-gathering system could distinguish between references to illicit drugs and medicinal drugs through the application of WSD. Research works such as named entity recognition and bioinformatics research can also use WSD. In the realm of information retrieval (IR), the primary concern lies in determining the accurate sense of a polysemous word within a given query before initiating the search for its corresponding answer [9,10]. Enhancing the efficiency and effectiveness of an IR system entails the resolution of ambiguity within a query. Similarly, in sentiment analysis, the elimination of ambiguity is crucial for determining the correct sentiment tags (e.g., negative or positive) associated with a sentence [11,12]. In question-answering (QA) systems, WSD assumes a significant role in identifying the appropriate types of answers that correspond to a given question type [13,14]. Furthermore, WSD is necessary to accurately assign the appropriate part of speech tagging (POS) to a word, as its POS can vary depending on the contextual usage [15,16].

WSD can be categorized into two classifications: "all words WSD" and "target word WSD". In the case of all words WSD, the disambiguation process extends to all the words present in a given sentence, whereas target word WSD specifically focuses on disambiguating the target word within the sentence. WSD poses a significant challenge within the field of NLP and remains an ongoing area of research. It is regarded as an open problem, categorized as "AI-Complete", signifying that a viable solution does exist but has not yet been discovered. If we consider the given below two sentences in the Hindi language

आज-कल बाज़ार में नई-नई वस्तुओं की माँग बढ़ रही है ।

(aaj-kal baazaar mein naee-naee vastuon kee maang badh rahee hai)
(Now-a-days the demand of new things is increasing in the market.)

सुहागन औरतें अपनी माँग में सिंदूर भरती हैं ।

(suhaagan auraten apanee maang mein sindoor bharatee hain)
(Married women apply vermillion on their maang (the partition of hair on head).)

In both sentences, we have a common word, "माँग" (maang), that has a different meaning as per the context. In the initial sentence, the term refers to "the demand", whereas in the subsequent sentence, it denotes "the partition of hair on the head". Identifying the specific interpretation of a polysemous word is not a problem for a personage, whereas, for machines, it is a challenging task. Conversely, Hindi is the top fourth language, with over 615 million speakers worldwide. A significant amount of work is performed for English

WSD, but the WSD for the Hindi language is still in its infancy stage. Hindi WSD is now gaining the attention of researchers.

The objective of this paper is to provide a comprehensive survey of the existing approaches and techniques for WSD in the Hindi language. It presents several approaches employed for WSD in the context of Hindi. The paper highlights the specific challenges and limitations faced in WSD for Hindi due to its morphological complexity, rich lexical resources, and less availability of labeled data. The rest of this paper is structured in the following way: Section 2 discusses the various approaches for WSD, followed by a proposed methodology presented on WSD in Section 3. In Section 4, the survey results presented for WSD were critically reviewed, and Section 5 is the conclusion.

2. Various Approaches for WSD

Various approaches and methods used for WSD are classified into two categories, including knowledge-based approaches and ML (Machine Learning) based approaches. In knowledge-driven approaches, external lexical resources such as Wordnet, dictionary, and thesauri are required to perform WSD, and in ML-based techniques, classifiers are trained to carry out the WSD task on sense-annotated corpora. Figure 2 presents the different WSD approaches, and the explanation for each category can be explained further.

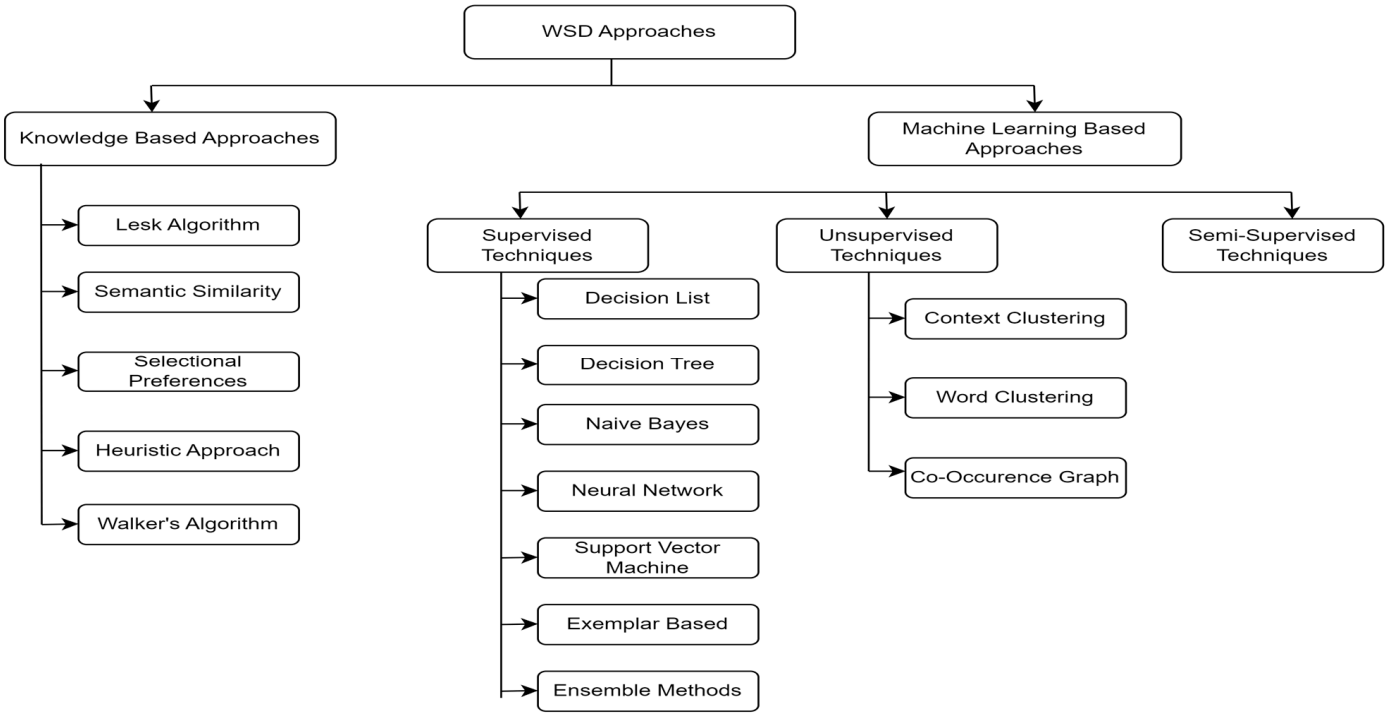

Figure 2. Classification of WSD Approaches.

2.1. Knowledge-Based Approaches

The knowledge-driven approach depends on various sources of knowledge such as dictionaries, thesaurus, ontologies, and collocations. The goal of these approaches in WSD is to utilize these knowledge resources to deduce the meanings of words within a given context. Let us delve deeper into an overview of several knowledge-based approaches.

- **LESK Algorithm**

The first algorithm developed using the knowledge-driven approach for WSD is the LESK algorithm [17,18]. The method relies on determining the degree of word overlap between the definitions or glosses of two or more target words. The dictionary definitions or glosses of the polysemous word are collected from the dictionary, and then these glosses and context words are compared. The desired sense of the polysemous word is determined

by identifying the sense with the highest degree of overlap. A score is calculated for each pair of word senses using the provided formula, which is

$$\text{overlapscoreLesk(S1, S2)} = |\text{Gloss(S1)} \cap \text{Gloss(S2)}|$$

The senses of the respective words are assigned based on the maximum value obtained from the above formula, where *Gloss(Si)* represents the collection of words in the textual interpretation of sense *Si* of word *W*.

- **Semantic Similarity**

Words that exhibit a connection with one another possess a shared context, allowing for the selection of the most suitable sense of a word by leveraging the meanings found within the shortest semantic distance. Various metrics can be employed to compute the semantic similarity between two words [19].

- **Selectional preferences**

Selectional preferences provide insights into the categories of words that are likely to be associated with one another and convey shared knowledge [20,21]. For instance, "actors" and "movies" are words that exhibit semantic relationships. In this approach, inappropriate senses of words are excluded, and only those senses that align with common sense rules are taken into consideration. The methodology revolves around tallying the occurrences of word pairs with syntactic relations in a given corpus. The identification of word senses is accomplished based on this frequency count.

- **Heuristic Approach**

In the heuristic approach, to disambiguate word heuristics, they are calculated using the different linguistic properties. Three types of heuristics are employed as a baseline.

(a) The most frequent sense heuristic operates on the principle of identifying all possible meanings that a word can have, with the understanding that one particular sense occurs more frequently than others.
(b) The one sense per discourse heuristic posits that a term or word maintains the same meaning throughout all instances within a specified text.
(c) The one sense per collocation heuristic has a similar meaning to the one sense per discourse heuristic, but it assumes that nearby words offer a robust and consistent indication of the contextual sense of a word.

- **Walker's Algorithm**

Walker introduced an approach or technique for WSD in 1987 [22,23]. This approach incorporates the use of a thesaurus to accomplish the task. The initial step involves assigning a thesaurus class to each sense of a polysemous word. Subsequently, a total sum is computed by considering the context where the ambiguous word appears. If the context of the word matches the word sense with a thesaurus category, the total sum for that category increases by one.

2.2. ML-Based Approaches

In ML-based approaches, a classifier undergoes a training step to acquire knowledge of the attributes and subsequently determines the senses for the unseen examples. The resources that are used in this approach are based on a corpus that can be tagged or untagged. In these types of approaches, the target is the word to be disambiguated, also called the input word, and the surrounding text in which it is submerged is referred to as the contextual information. ML-based approaches are categorized into three types: supervised, unsupervised, and semi-supervised techniques.

2.2.1. Supervised Techniques

Supervised techniques for disambiguation utilize sense-annotated corpora for training purposes. These techniques operate under the supposition that the context itself can impart

sufficient affirmation to resolve a sense of ambiguity. The context is represented as a collection of word "features", encompassing information about the neighboring words as well. Within these techniques, a classifier is trained using a designated training set that consists of instances specifically related to the target word. Overall, supervised approaches in WSD have generally achieved superior results compared to other approaches. However, the problem is that these techniques work on sensing annotated dataset, which is very expensive to create. Various supervised techniques are as follows:

- **Decision list**

In the context of WSD, a decision list refers to a sequential collection of "if-then-else" rules that are employed to determine the suitable sense for a given word [24,25]. It can also be viewed as a listing of weighed "if-then-else" rules. These rules are generated from a training set, utilizing parameters such as feature value, sensitivity, and score. The decision list is constructed by arranging these rules in descending order of their scores. When encountering a word, let us say w, its frequency of existence is computed, and its representation as a feature vector is used to evaluate the decision list, resulting in a calculated value. The attribute that has the highest value that matches the input vector corresponds to the meaning assigned to the word w.

- **Decision Tree**

A decision tree is a classification method that repeatedly divides the training dataset and organizes the classification rules in a tree-like structure [26,27]. Every interior node of the decision tree represents a test performed on an attribute value, and the branches represent the outcomes of the test. The word sense is determined when a leaf node is reached. An illustration of a decision tree for WSD is depicted in Figure 3. In this example, the sense of the polysemous word "bank" that is active is a noun within the sentence, "I will be at the bank of the Narmada River in the afternoon." The tree has been constructed and traversed to ultimately select the sense "bank/RIVER." A null value in a leaf node indicates that there is no sense selection present for that particular attribute value.

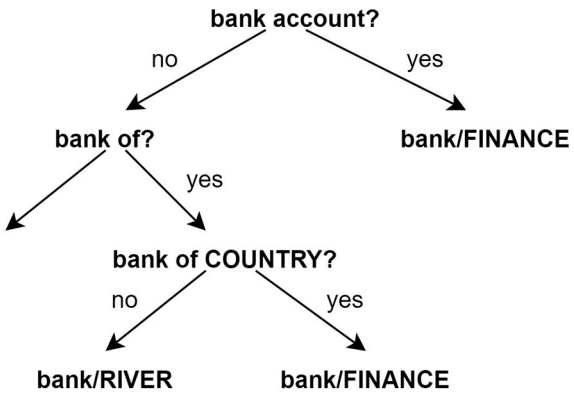

Figure 3. Decision Tree Example.

- **Naïve Bayes**

The NB (Naïve Bayes) classifier is a probabilistic classifier that applies Bayes' Theorem [28,29] to determine the appropriate meaning for a word. To classify text documents, it computes the conditional probability of each sense *Si* of word w based on the context features j. The sense S with the highest value, determined using the provided formula, is chosen as the most appropriate sense within the given context.

$$\hat{S} = \underset{S_i \in Sense_D(w)}{\operatorname{argmax}} P(S_i \mid f_1, \ldots, f_m) = \underset{S_i \in Senses_D(w)}{\operatorname{argmax}} \frac{P(f_1,\ldots, f_m|S_i)P(S_i)}{P(f_1,\ldots, f_m)}$$
$$= \underset{S_i \in Senses_D(w)}{\operatorname{argmax}} P(S_i) \prod_{j=1}^{m} P(f_j|S_i)$$

In this context, m denotes the number of features. The probability score $P(Si)$ is computed based on the co-existence frequency of senses in the training dataset, while $P(fj \mid Si)$ is derived using the presence of the attribute given in the sense.

- **Neural network**

Neural networks consist of interconnected units or artificial neurons that serve as a loose model of human brain neurons [30,31]. They follow a connectionist approach and utilize a computational model for data processing. The learning program receives input attributes and target output. The objective is to divide the training data into non-overlapping sets based on desired responses. When new input pairs are presented to the network, the weights are adjusted to ensure the higher activation of the output unit that generates the desired result compared to other output units. In the context of neural networks, nodes represent words, and these words activate the associated concept with which they share semantic relations. Inputs propagate from the input to the output layer through intermediate layers. The network efficiently processes and manipulates the inputs to generate an output. However, generating a precise output becomes challenging when the connections within the network are widely dispersed and form loops in multiple directions.

- **Support Vector Machine (SVM)**

An SVM [32] serves the purpose of both classification and regression tasks. This approach is rooted in the concept of identifying a hyperplane that can effectively isolate positive examples from negative ones with the highest possible margin. The edge/margin represents the interspace between the hyperplane and the nearest examples for positive and negative, which are referred to as support vectors. In Figure 4, circle and square represent two different classes, the bold line represents the hyperplane that isolates the two classes while the dashed lines indicate the support vectors closest to positive and negative example. These support vectors play an important role in constructing an SVM classifier. The vectors have an impact on the position and the orientation of the hyperplane, and by removing or adding support vectors, adjustments can be made to the position of the hyperplane. In Figure 4,

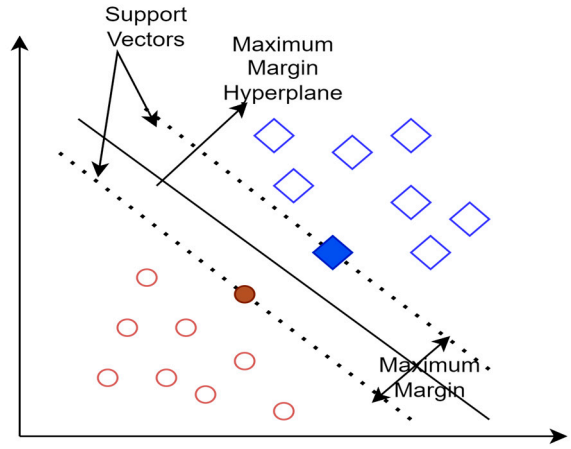

Figure 4. Illustrating SVM Classification.

- **Exemplar or instance-based learning**

In this approach, the classification model is constructed using examples [33]. In a feature space, these examples are represented as points, and the new examples are evaluated for classification. When new examples are encountered, they are progressively stored in the model. The k-nearest neighbor (k-NN) [34] method is an example of this type of approach. In k-NN, examples are stored based on their feature values, and the classification of the new examples is determined by considering the meanings of the k most similar previously stored examples. The hamming distance (a measure of the number of differing elements

between two strings of equal length) [35,36] is calculated between new examples and the stored examples using the k-NN algorithm, which measures the proximity of the given input to the stored examples. The highest value obtained among the k-nearest neighbors represents the output sense.

- **Ensemble methods**

In order to enhance the accuracy of disambiguation, it is common to employ a combination of different classifiers. This combination strategy is called ensemble methods, which combine algorithms of different nature or with different characteristics [37]. Ensemble methods are more powerful than single-supervised techniques as they can overcome the weakness of a single approach. Strategies such as majority voting, the AdaBoost system of Freund and Schapire [38], rank-based combination, and probability mixture can be utilized to combine the different classifiers to improve accuracy. Figure 5 presents the simple approach of the ensemble WSD approach.

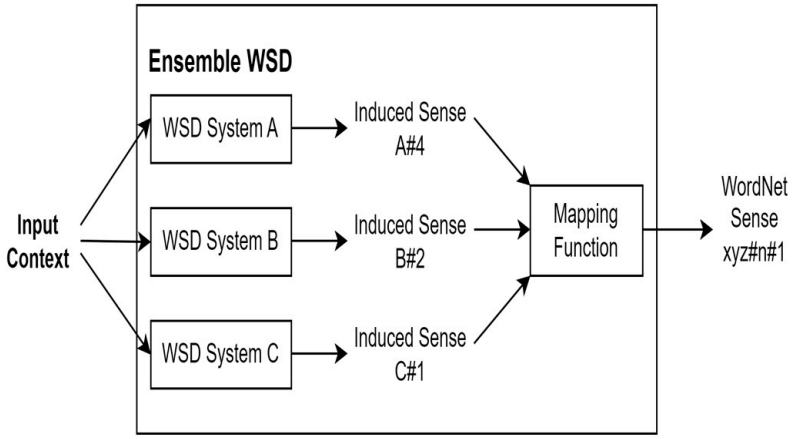

Figure 5. Ensemble Methods: Combining the Strengths of Multiple Models.

2.2.2. Unsupervised Techniques

Unsupervised techniques do not make use of sense annotated datasets or external knowledge sources. Instead, they operate under the assumption that senses with similar meanings occur in similar contexts. These techniques aim to determine senses from the text by clustering the word occurrences based on some measure of contextual similarity. This task is known as word sense induction or discrimination. Unsupervised techniques offer significant potential in overcoming the bottleneck of knowledge acquisition, as they do not require manual efforts. Here are some approaches that are used for unsupervised WSD.

Context Clustering: This unsupervised approach is rooted in the use of clustering techniques [39]. It begins by representing words through context vectors, which are then organized into clusters. Each cluster is corresponding to a sense of the target word. The approach revolves around the notion of a word space or vector space, where the dimensions represent individual words. Specifically, a word w is transformed into a vector, capturing the frequency of its co-occurrences with other words. This leads to the creation of a co-occurrence matrix, which is then subjected to various similarity measures. Finally, sense discrimination is performed by applying clustering techniques such as k-means clustering or agglomerative clustering.

Word Clustering: The induction of word senses can also be achieved through the use of word clustering [3]. This approach groups words that are semantically similar and may possess specific meanings. One commonly employed method for word clustering is Lin's method [40], which identifies words that are synonymous or have similarities to the target word. The similarity among the synonyms and the target word is determined by analyzing the features represented by syntactic dependencies found in a corpus, such as a verb–object, subject–verb, adjective–noun relationships, and so on. The more similar the two words are,

the greater the extent to which they share information content. A word clustering algorithm is then utilized to differentiate between senses. Given a list of words W, the words are initially arranged based on their similarity, and a tree for similarity is constructed. In the beginning, the tree has only a single node, and through iterations, the most similar word is added as a child to the tree for each word in the list. Subsequently, pruning is performed, resulting in the generation of sub-trees. Each sub-tree, with the initial node serving as its root, represents a distinct sense of the original word.

Another method that is used for the clustering of words is the clustering by committee (CBC) [41] algorithm. The first step is similar to the above, i.e., a set of similar words is created for each input word. A similarity matrix is constructed to capture the pairwise similarity information between words. The second step involves the application of a recursive function to determine a set of clusters, referred to as committees. Following this, the average-link clustering technique is applied. In the final step, a discrimination process is executed, assigning the most alike cluster to each target word according to its similarity to the centroid of each committee. Subsequently, the intersecting attributes among the word and the committee are eliminated from the initial/actual word. This allows for the identification of less frequent senses for the same word in the next iteration.

Co-occurrence Graph: This approach utilizes a graph-based methodology. It involves the creation of a co-occurrence graph [42], denoted as G, comprising vertices V and edges E. Words are represented as vertices, and the connections between words that co-occur within the same paragraph are represented as edges. The weight assigned to each edge is determined by the frequency of co-occurrences, thus capturing the relationships between connected words. This graph construction effectively portrays the grammatical relations between the words.

In order to determine the sense of a word, an iterative method is used to identify the word with the highest degree node in the graph. Subsequently, a minimum spanning tree algorithm is applied to deduce the word's sense based on the information extracted from the graph. This process allows for a meaningful sense of disambiguation of the word within the given context.

2.2.3. Semi-Supervised Techniques

Semi-supervised techniques, known as weakly supervised or minimally supervised approaches, are utilized in WSD when training data are scarce. These methods make efficient use of both labeled and unlabeled data. Among the earliest algorithms in the realm of semi-supervised approaches is bootstrapping. Bootstrapping involves statistical resampling, where multiple datasets are generated from the original data with replacement. This technique is employed to estimate the accuracy and variability of a model or statistical inference, particularly in cases where traditional assumptions are not applicable or when working with small datasets.

The following table, Table 1 gives an in-depth comparison of various WSD approaches based on their benefits, drawbacks, and rationale for use. It seeks to provide a thorough grasp of how each method works and the settings in which they excel or may have limits.

Table 1. Comparative Analysis of Knowledge-Based, Supervised, Unsupervised, and Semi-Supervised Techniques.

Technique	Working	Advantages	Disadvantages	Justification for Usage
Knowledge-based	Utilizes pre-defined rules and human expertise to make decisions or classify data.	1. Interpretable outcomes	1. Limited scalability	Useful when domain-specific knowledge is available and interpretability is essential
		2. Robust to noisy data	2. Relies on expert knowledge	

Table 1. *Cont.*

Technique	Working	Advantages	Disadvantages	Justification for Usage
Supervised	Trained on labeled data with input–output pairs and predicts outputs for unseen data based on the learned model.	1. High accuracy	1. Requires labeled data	Preferred when labeled data are available and the goal is precise predictions
		2. Well-established algorithms	2. Sensitive to outliers and noise	
		3. Suitable for various problem types (classification, regression, etc.)	3. Lack of generalization to unseen classes or categories	
Unsupervised	Clusters data or discovers hidden patterns without labels.	1. Useful for exploratory data analysis	1. Limited guidance in model evaluation	Ideal for identifying structures in data when labeled data are scarce or unavailable.
		2. Can handle large datasets	2. Lack of direct feedback on model performance	
		3. Detects anomalies or outliers	3. Difficulty in interpreting the results	
Semi-supervised	Utilizes a combination of labeled and unlabeled data.	1. Utilizes the advantages of both supervised and unsupervised learning	1. Difficulty in obtaining and managing labeled data	Valuable when labeled data are expensive to acquire but unlabeled data are abundant
		2. Cost-effective for certain applications	2. Semi-supervised methods may not outperform fully supervised or unsupervised techniques	
		3. Improves performance with limited labeled data	3. May suffer from error propagation due to incorrect labels	

3. WSD Execution Process

WSD is the task of determining an ambiguous word's suitable sense based on context. WSD has seen a variety of methods. The majority of methods are based on different statistical methods. A few methods use corpora that have been sense-tagged, while others use unsupervised learning. The flowchart in Figure 6 shows the steps that are performed for WSD.

A string with an ambiguous word is given as an input string. Then, pre-processing is performed on this input string. Pre-processing steps such as stop word elimination, tokenization, part-of-speech tagging, and lemmatization, etc., are essential to transform raw text into a suitable format for analysis. For example, we have input 'राम कच्चा आम खा रहा है।' (raam kachcha aam kha raha hai) (Ram is eating raw mango). Various pre-processing steps are as follows:

Stop Word Elimination: Stop words are words commonly filtered out or excluded from the analysis process in NLP. These words are highly frequent in most texts, but they generally lack significant meaning or do not contribute much to the overall understanding of the content. By eliminating stop words, the text becomes less noisy, and contextual relevance is improved. This improved context helps the WSD algorithm make more accurate sense selections.

Examples of stop words in English include "the", "a", "an", "in", "on", "at", "and", "but", "or", "I", "you", "he", "she", "it", etc. Examples of stop words in Hindi

(Devanagari script) include "का," "की," "के," "को," "है," "हैं," "में," "और," "लेकिन," "या," "मैं," "तुम," "वह," "यह," आदि।.

The elimination of stop words and punctuation from the input text is performed in this step as they hold no significance or utility. After stop word removal string is 'राम कच्चा आम खा'.

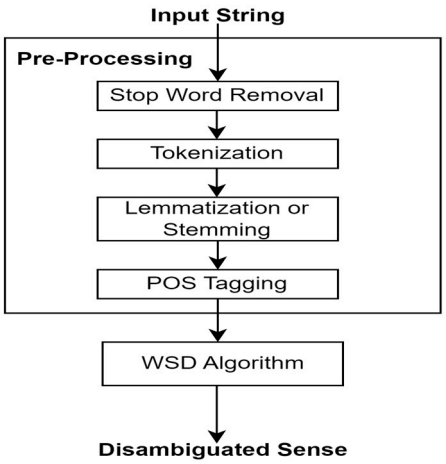

Figure 6. Flowchart of WSD Execution Process.

Tokenization: Tokenization is a fundamental technique in NLP that involves dividing a given text into smaller components, such as sentences and words. It encompasses the method of breaking down a string into a list of individual units called tokens. It helps in isolating individual words for disambiguation, making the WSD process more manageable and focused. In this context, a token can refer to a word within a sentence or a sentence within a paragraph, representing a fragment of the whole text. After Tokenization output is ('राम', 'कच्चा', 'आम', 'खा').

Stemming: Stemming is a linguistic process aimed at removing the last few characters of a word, which can sometimes result in incorrect meanings and altered spellings. Stemming simplifies text data and improves computational efficiency, aiding in tasks such as text matching and retrieval. However, it may generate non-words, leading to potential loss of word meaning and semantic distinctions. For instance, stemming the word 'Caring' would return to 'Car', which is an incorrect result.

Lemmatization: Lemmatization takes into account the context of a word and transforms it into its meaningful base form, known as a lemma. For example, by lemmatizing the word 'Caring,' the resulting lemma would be 'Care', which is the correct result. By converting words to their lemma, the WSD system can associate different inflected forms of a word with the same sense, improving the coverage and generalization of the sense inventory.

PoS Tagging: POS tagging involves the assignment of suitable part-of-speech labels to each word within a sentence, encompassing categories such as nouns, adverbs, verbs, pronouns, adjectives, conjunctions, and their respective sub-categories. This information is crucial for WSD because different parts of speech may have different senses. POS tagging helps in narrowing down the sense options for each word based on its grammatical role in the sentence.

When pre-processing is completed, the WSD algorithm is applied that gives the accurate sense of the ambiguous word as output. Various WSD algorithms are supervised, semi-supervised, unsupervised, and knowledge-based.

WordNet [43] is a valuable tool that plays a significant role in WSD. It serves as an extensive database containing nouns, adjectives, verbs, and adverbs, which are arranged into clusters of synonymous word groups known as synsets. These collections are interconnected through applied lexical and semantic relations. At IIT Bombay, Hindi WordNet

(HWN) [44] is being developed, which shares similarities with English WordNet. Words are grouped based on their perceived similarity in impact in HWN. It is worth noting that in certain contexts, terms that may have distinct meanings elsewhere can be considered synonymous. Each word within the HWN is associated with a corresponding synset that stands for "synonym set" and represents a group of words or terms that are synonymous or have similar meanings representing a lexical concept.

The WordNet synsets serve as its primary construction blocks. HWN controls words with open class categories or words with substance. Thus, the noun, adjective, verb, and adverb word categories that make up the HWN are included. The following characteristics apply to every entry in the HWN:

- Synset: This is a group of words, or synonyms, with similar meanings. For example, "पेन, कलम, लेखनी" (pen, kalam, lekhanee) refers to a tool or device used for writing with ink. According to the frequency of usage, the words are organized in the synset.
- Gloss: It explains the ideas. It is divided into two sections: a text definition that explains the concepts indicated by the synset (for example, "स्याही के सहयोग से कागज आदि पर लिखने का उपकरण (syaahee ke sahayog se kaagaj aadi par likhane ka upakaran)" elaborates on the idea of a writing or drawing instrument that utilizes ink), along with an illustrative sentence showcasing the importance of each word within a sentence. In general, a synset's words may be simply changed in a phrase (for instance, "यह पेन किसी ने मुझे उपहार में प्रदान की है। (yah pen kisee ne mujhe upahaar mein pradaan kee hai) (Someone gifted me this pen.)" illustrates the usefulness of the synset's words describing an ink writing or drawing equipment).

4. Results and Discussions

In this section, we presented the overview of which techniques and methodologies have been used by different researchers and what accuracy they have achieved, which datasets have been used by them, and what is specific about their techniques. We have divided it according to the techniques used by different researchers. It will help the researchers in the future to analyze which technique they should use.

4.1. Knowledge-Based Techniques

Knowledge-based techniques for WSD rely on external knowledge resources to resolve word ambiguities. These techniques use lexical databases, semantic networks, and linguistic resources to associate words with their appropriate meanings based on contextual information. As researchers delved into the subject, they started employing a combination of automatic knowledge extraction techniques alongside manual methods. Various knowledge-based techniques used by researchers for WSD are as follows:

In 1986, the first algorithm, called the Lesk algorithm [18], was developed by Michael Lesk for the disambiguation of words. In this algorithm, overlapping of the context where the word occurs and the definition of the input word from the Oxford Dictionary (OALD) was performed. The sense with the maximum overlap is chosen as the correct sense of the ambiguous word. In [17], Banerjee and Pederson introduced an adapted Lesk approach that relied on utilizing a lexical database, WordNet, as a source of knowledge rather than a machine-readable dictionary. WordNet, a hierarchical structure of semantic relations such as synonyms, hypernyms, meronyms, and antonyms, served as the foundation for this algorithm.

The notion of disambiguating Indian languages was initially proposed with a technique involving a comparison of contexts within which ambiguous words occurred with those created with HWN [45]. The sense would be determined according to its degree and extent of overlap. HWN arranges the lexical information based on word meanings. Hindi WordNet's design was influenced by English WordNet. HWN was developed by IIT Bombay, and it became publicly available in 2006. The accuracy range is about 40% to 70%.

Singh et al. [46] investigated the impact of the size of context window, stemming, and stop word removal on the Lesk-like algorithm for WSD in Hindi. The elimination of stop

words coupled with the use of stemming is a proven method for obtaining good results, and they applied the Lesk algorithm to their work. From the analysis carried out, it is evident that utilizing 'Karak relations' leads to correct disambiguation. Additionally, stop-word elimination combined with stemming can help to raise the number of content-specific vocabulary while also promoting greater word stem overlap. A 9.24% improvement in precision is reported after the elimination of stop words and stemming over the baseline. In [47], the WSD technique relies on graph-based methods. They merged Lesk semantic similarity measures with Indegree approaches for graph centrality in their study. The beginning step involves constructing a graph for all target words in the sentence wherein nodes correspond to words and edges denote their respective semantic relations. By using Hindi wordNet along with the DFS Algorithm, we managed to create a final graph. The determination of a word's meaning is ultimately achieved through the application of graph centrality measures. An accuracy rate of 65.17% is achieved.

The authors introduced and evaluated the effectiveness of Leacock–Chodorow's measure of semantic relatedness for WSD of Hindi [48]. Having semantic similarity between two terms indicates a relationship. Semantic similarity and additional relations such as is-a-kind-of, is-the-opposite-of, is-a-specific-example-of, is-a-part-of, etc., are included in the relationships between ideas. The Leacock–Chodorow metric is employed, taking into account the length of routes among the noun concepts within an is-a hierarchy. The algorithm employs the Hindi WordNet hierarchy to acquire word meanings and uses it in the process of disambiguation rather than relying solely on the direct overlap. For evaluation purposes, a dataset consisting of 20 sense-tagged polysemous Hindi nouns is utilized. Using this metric, they found an accuracy of 60.65%. The role of hypernym, holonym, meronym, and hyponym interactions in Hindi WSD is examined [49]. We have taken into account five different scenarios in their research, including all relations, hyponym and hypernym, hypernym, holonym, and hyponym. The baseline makes no use of any semantic relations. When taking into account all relations, they found that total precision had increased by 12.09% over the baseline. The use of hyponyms produced the greatest improvement for a single semantic link and a precision improvement of 9.86% overall.

Sawhney et al. [50] employed a modified Lesk approach that incorporates a dynamic context window concept. The dynamic context window refers to the number of preceding and succeeding words surrounding the ambiguous words. According to this approach, if two words have similar meanings, then there must be a common topic in their vicinity. An increase in precision signifies that this algorithm provides superior results as compared to prior methods that employ a fixed-size context window. The lesk approach was applied to bigram and trigram words to disambiguate the verb words [51], and it is the only work, as per our knowledge, that disambiguates Hindi verbs, as most of the work is performed for nouns.

In [52], Goonjan et al. make use of Hindi Wordnet to retrieve the senses of the words, and then a graph is created using a depth-first search between the senses of the words. After that, weights are assigned to the edges of the connecting node according to the weights of the Fuzzy Hindi wordnet. Then, various local fuzzy centrality measures are applied, and the values of these calculated measures help us to find the accurate meaning of the polysemous word. The knowledge-driven Lesk algorithm is employed in [53] that works by selecting the meaning whose definition most closely matches the In their investigation, they successfully identified 2143 out of 3000 ambiguous statements, achieving an accuracy rate of 71.43%.

In [54], WSD for the Bengali language is performed in two distinct phases. During the first phase, sense clusters of an ambiguous word are constructed by considering the preceding and succeeding words in their context. In the second phase, WSD is performed by utilizing a semantic similarity measure after expanding the context with the assistance of Bengali WordNet. An ambiguous Bengali words test set, comprising 10 words, is used, for testing which has 200 sentences for each ambiguous word. The overall accuracy achieved is 63.71%. Tripathi et al. [55] have used a Lesk algorithm along with a novel scoring method.

To enhance the performance of the Lesk Algorithm, they employed a scoring technique that evaluates token senses based on their cohesive variations. This strategy aimed to improve the accuracy and effectiveness of the approach. Based on a combination of different senses of tokens according to the gloss along with their related hypernyms and hyponyms, a sense rating is assigned that helps in determining the meaning of the ambiguous word.

A complete framework named "hindiwsd" [56] is constructed for WSD of Hindi in Python language. It is a pipeline that performs a series of tasks, including transliteration of Hinglish code-mixed text, spell correction, POS tagging, and the disambiguation of Hindi text. A knowledge-based modified Lesk algorithm is applied here for WSD. A comparative analysis of various knowledge-based approaches is also performed in [57]. The results demonstrate that accuracy is lower for limited resource languages and higher for languages with abundant knowledge resources. A knowledge-based resource is critical in the processing of any language. The survey suggests that several factors influence the performance of WSD tasks. These include the removal of stop words, the positioning of ambiguous words, the use of Part-of-Speech (POS) tagging, and the size of the dataset utilized for training. Each of these elements plays a significant role in the overall effectiveness of WSD methods.

This is a review of some knowledge-based approaches that have been used by different researchers for WSD. Knowledge-based techniques can be effective in resolving word sense ambiguities, especially when supported by comprehensive and well-structured lexical resources and linguistic knowledge. However, they may have limitations when dealing with unseen or domain-specific contexts, as they heavily rely on the information present in the knowledge bases. In such cases, supervised and unsupervised machine learning approaches are often employed to complement the knowledge-based methods and improve overall disambiguation performance.

4.2. Supervised Techniques

Supervised techniques for WSD are highly effective in resolving word sense ambiguities by utilizing labeled training data, achieving high accuracy through diverse and well-annotated datasets that associate words with correct senses in various contexts. These methods capture deeper semantic relationships, enabling a nuanced understanding of word sense distinctions while exhibiting context sensitivity to handle complex sentence structures and resolve ambiguous words. We present a review of various supervised techniques used for WSD of Indian languages.

NB classifier [58], a supervised method equipped with eleven different features such as collocations, vibhaktis vibhaktis (the grammatical cases or inflections used in Indian languages to indicate the function of nouns or pronouns in a sentence), unordered list of words, local context, and nouns has been applied to solve Hindi WSD. In order to assess its performance, the NB classifier was applied to a set of 60 polysemous Hindi nouns. Applying morphology to nouns included in a feature vector led to achieving maximum precision of 86.11%, while considering the nearby nouns in the context of a target ambiguous noun is important for achieving accurate meaning.

In [59], a supervised approach using cosine similarity is introduced. Vectors have been generated for the query given for testing and knowledge data for the sense of the polysemous word, taking weights into account. The sense with the maximum similarity to the polysemous word is selected as the appropriate sense. The experiment is conducted on a dataset comprising 90 Hindi-ambiguous words. An average precision of 78.99% is obtained.

The supervised approach of the k-NN algorithm has been used for Gurumukhi WSD [60]. Two feature sets are derived: one comprises frequently occurring words alongside the ambiguous word, and the other encompasses words neighboring the ambiguous word in the corpora. Subsequently, the provided data are divided into the training and the testing sets. The k-NN classifier is trained using the training set. For the given input sentence, pre-processing is performed, and then its vector is generated. The k-NN classifier identifies similar vectors or nearest neighbors for the unknown vector. After that, the

distance between the input vector/unknown vector and nearest neighbors is calculated using the Euclidean method. The closeness between the vectors is determined by using this distance.

The WSD of Panjabi has been accomplished using a supervised NB [61] classifier. For feature extraction, both Bag-of-Words (BoW) and a collocation model are employed. The collocation model utilizes only the two preceding and two succeeding words of the input word as features, whereas the BoW model considers all the words surrounding the input word as features. Using both feature extraction methods, the NB classifier is trained on a dataset of 150 ambiguous words with six or more senses collected from the Punjabi word net. The system attains an accuracy of 89% for the Bow model, and for the collocation model, the accuracy is 81%.

In [62], a comparative analysis is conducted among rule-based, classical machine learning, and two neural network-based RNN and LSTM models. The evaluation is carried out on four highly ambiguous terms and a group of seven other ambiguous words. The rule-based method achieved an accuracy of 31.2%, classical machine learning attained 33.67% accuracy, while RNN exhibited an accuracy of 41.61%. Notably, the LSTM model outperformed all other methods with an accuracy of 43.21%, showcasing its superior performance in disambiguating word senses.

A review of some supervised techniques is presented. Supervised techniques excel in providing fine-grained disambiguation, which is essential for precise semantic interpretation. However, their dependency on labeled data poses challenges, especially for resource-limited languages. Supervised techniques may struggle with unseen words or senses, and overfitting remains a concern, potentially affecting performance on new data. To address limitations, researchers often combine supervised methods with unsupervised or knowledge-based approaches to enhance overall WSD performance.

4.3. Unsupervised Techniques

Unsupervised techniques for WSD present advantages in their independence from labeled training data, making them more cost-effective and adaptable to different languages and domains. By learning solely from distributional patterns, they have the potential to discover new word senses and uncover novel semantic relationships. A review of unsupervised techniques used for WSD of Indian languages is as follows:

An unsupervised approach is used for resolving word ambiguity in [63]. As part of the pre-processing steps, the elimination of stop words and stemming is required when encountering an unclear context. After employing the decision list for untagged examples, there is a need for some manual intervention to provide seed examples. A decision list is employed to generate ambiguous words, and this decision list is subsequently utilized to determine the sense of such ambiguous words.

A technique to perform unsupervised WSD on a Hindi sentence using network agglomeration is proposed in [64]. We start by creating a graph G for the input sentence. All variations in meaning for this sentence can be seen collectively in this graph. Sentence graphs can be used to develop interpretation graphs such as G, and the sentence must have an interpretation for all instances of G. To find out which is the preferred interpretation, we perform network agglomeration on all relevant graphs. By identifying which interpretation holds the highest network agglomeration value, we can derive its relevance.

In [65], the author deals with algorithms based on an unsupervised graph-based approach. This consists of two phases: (1) A lexical knowledge base is utilized to construct a graph, where each node and edge in the graph represents a possible meaning of a word within a given sentence. These nodes and edges capture dependencies between meanings, such as synonyms and antonyms. (2) Subsequently, the graph is analyzed to identify the most suitable node, representing the most significant meaning, for each word according to the given context. In the graph-based WSD method of unsupervised techniques, word meanings are determined by considering the dependencies between these meanings.

Relations in HWN are crisp, meaning they are either related or not related at all. There is no partial relation between words in the Hindi wordnet. However, in real life, partial relations can also exist between words, which are also called fuzzification of relations. Therefore, an expanded version of Hindi wordnet that incorporates fuzzy relations is called Fuzzy Hindi WordNet (FHWN), which is represented as a fuzzy graph in which nodes depict words/synsets and the edges show fuzzy relationships within words/synsets. The fuzzy relations are assigned a membership value between [0, 1]. The values are assigned by consulting with experts from diverse domains. In [66], an approach using fuzzy graph connectivity measures is applied to FHWN for WSD. Various local and global connectivity measures are calculated using the values assigned to the relations. The sense with the maximum rank is chosen as the suitable sense for the ambiguous word. The utilization of the FHWN sense inventory results in an improvement in disambiguation performance, with an average increase of approximately 8% in most cases. Since the membership value can change, so can the algorithm's performance.

In [67], a multilingual knowledge base called ConceptNet is used to automatically generate the membership values. The nodes and edges that make up ConceptNet's network represent words, word senses, and brief phrases, while the edges show how the nodes are related to one another. The Shapley value, which is derived from co-operative game theory, is then employed as a centrality metric. Shapley's value is utilized to mitigate the influence of alterations in membership values within fuzzy relationships by considering only the marginal contributions of all the values in the calculation of centrality.

For Gujarati WSD [68], a genetic algorithm-based strategy was employed. Darwin's idea of evolution serves as the basis for genetic algorithms. The population is the first set of solutions the algorithm considers (represented by chromosomes). One population's solutions are utilized to create a new one. This approach is pursued with the expectation that the new population will exhibit improved performance compared to the previous population. The solutions chosen to create new descendants (solutions) are selected based on their suitability. This process is carried out again and again until or unless a certain need (such as the number of people or an improvement in the ideal solution) is attained.

Kumari and Lobiyal [69] introduced a word-embedding-based approach for WSD. They employed two word2vec architectures, namely the skip-gram and the continuous bag-of-words models, to generate word embeddings. The determination of the appropriate sense of a word was achieved using cosine similarity. An unsupervised Latent Dirichlet Allocation (LDA) and Semantic features-based approach using semantic features has been applied for the target WSD of the Malayalam language [70]. A dataset consisting of 1147 contexts containing target polysemous words has been utilized. In total, 80% accuracy is achieved.

Various word embedding methods such as Bow, Word2Vec, TF-IDF, and FastText have been used in [71]. For the construction of Hindi word embeddings, Wikipedia articles were used as the data source. They conducted multiple trials to explore this idea, and the results convinced us that Word2Vec outperforms all other embeddings for the Hindi dataset we examined. When training the input, the method uses word embedding techniques. It also incorporates clustering, which is used to create a sense inventory that aids in disambiguation. These methods can use unlabeled data because they are unsupervised. The accuracy achieved is 71%.

In [72], The authors employed an approach based on a genetic algorithm (GA) for Hindi WSD. The process involved pre-processing and creation of a context bag and sense bag, followed by the application of the GA. The GA encompassed selection, crossover, and mutation to disambiguate the word, and the approach was tested on a manually created dataset. The experimental results demonstrated an accuracy of 80%. A comparative analysis of two path-based similarity measures is performed in [73]. The experimental investigation is performed using the shortest path and Leacock–Chodorow methods, which shows that a Leacock–Chodorow similarity measure performs better than the shortest

path measure. Experimentation is performed on five polysemous nouns, and an average accuracy of 72.09% is achieved with the Leacock–Chodorow method.

Unsupervised techniques are cost-effective, and they use unlabeled data. Thus, they can be used for languages that lack sense-tagged datasets. However, they may struggle with sense overlapping and lack deep semantic interpretation, leading to less precise disambiguation compared to supervised methods. Data sparsity can also limit their effectiveness, requiring substantial data for satisfactory performance. Evaluating their performance can be challenging without a definitive gold standard for comparison. Combining unsupervised techniques with supervised or knowledge-based approaches can address their limitations and enhance overall WSD performance.

The following table, Table 2, exhibits the summary of study characteristics of different Indian language WSD approaches.

Table 2. Analysis of WSD Approaches in Different Indian Languages.

Year (Ref.)	Language	Technique	Method	Specification	Dataset Used	Accuracy	Comments
1986 [18]	English	Knowledge-Based	Lesk	Overlapping of context and word definition is performed.	Used Machine Readable Dictionaries	-	Only definitions are used for deriving the meaning.
2002 [17]	English	Knowledge-Based	Adapted Lesk	The proposed approach expands the comparisons by incorporating the glosses of words that are linked to the words under disambiguation in the given text. These connections are established using the WordNet lexical database.	WordNet is used	32%	-
2004 [45]	Hindi	Knowledge-Based	Lesk Method	Comparison of the ambiguous word's context and the context derived from Hindi WordNet is performed.	The manually created test set.	40–70%	Works with only nouns and does not deal with morphology.
2009 [63]	Hindi	Unsupervised	Decision List	After pre-processing, a decision list of untagged examples is created that is utilized to depict the meaning of the polysemous word.	A dataset for 20 ambiguous words with 1856 training instances and 1641 test instances was used.	The accuracy ranges from approximately 82% to around 92% when employing techniques such as stop-word elimination, automatic generation of decision lists, and stemming.	-

Table 2. *Cont.*

Year (Ref.)	Language	Technique	Method	Specification	Dataset Used	Accuracy	Comments
2012 [46]	Hindi	Knowledge-Based	Lesk Algorithm	Effects of context window size, stop word elimination, and stemming has been analyzed with Lesk	Evaluation is carried out on a test set of 10 polysemous with 1248 test instances.	Improvement of 9.24% over baseline.	Works only for nouns.
2012 [47]	Hindi	Knowledge-based	Graph-Based	A graph is constructed using the DFS algorithm and then centrality measures are applied to deduce the sense of the word.	Text files that contain 913 nouns are used as datasets.	65.17%	For graph centrality, only the in-degree algorithm is used.
2013 [48]	Hindi	Knowledge-Based	A Leacock-Chodorow measure of semantic relatedness	The Leacock–Chodorow algorithm is used to find the length of the route among two noun concepts.	dataset of 20 polysemous Hindi nouns	60.65%	Works only for nouns
2014 [49]	Hindi	Knowledge-Based	Semantic Relations	The significance of different relationships such as hypernym, hyponym, holonym, and meronym is examined here.	dataset of 60 nouns is used.	Improvement of 9.86% over baseline.	Only for nouns.
2014 [58]	Hindi	Supervised	Naive Bayes	Naive Bayes classifier with eleven different features has been applied for Hindi WSD.	A dataset of 60 polysemous Hindi nouns is used.	86.11%	Works only for nouns
2014 [50]	Hindi	Knowledge-Based	Modified Lesk	A modified Lesk approach with a dynamic context window is used in this paper.	A dataset of 10 ambiguous words is used.	-	Accuracy depends on the size of the dynamic context window.
2015 [64]	Hindi	Unsupervised	Network Agglomeration	An interpretation graph is created for each interpretation derived from the graph of the sentence, and subsequently, network agglomeration is performed to determine the correct interpretation.	Health and Tourism datasets are used.	Health-43% (All words) and 50% (Nouns) Tourism-44% (All Words) and 53% (Nouns)	Works for nouns as well as other parts of speech, too.

Table 2. *Cont.*

Year (Ref.)	Language	Technique	Method	Specification	Dataset Used	Accuracy	Comments
2015 [65]	Hindi	Unsupervised	Graph Connectivity	A graph is generated to represent all the senses of a polysemous word, then it is analyzed to determine the accurate sense of the word.	Hindi Wordnet is used as a reference library.	-	No standard dataset.
2015 [66]	Hindi	Unsupervised	Fuzzy Graph Connectivity Measures	Different global and local fuzzy graph connectivity measures are computed to find the meaning of a polysemous word.	Used Health corpus.	Performance increases by 8% when we use Fuzzy Hindi WordNet.	-
2016 [51]	Hindi	Knowledge-Based	Tri-Gram and Bi-Gram	Lesk's approach is applied to tri-gram and bi-gram verb words.	15 words of verbs are used as a dataset with 103 test instances.	52.98% with bi-gram and 33.17% with tri-gram.	Only work for verb words.
2016 [59]	Hindi	Supervised	Cosine Similarity	The cosine similarity of vectors, created from input query and senses from Wordnet, is calculated to determine the meaning of the word.	dataset of 90 Hindi ambiguous word	78.99%	It does not perform part-of-speech disambiguation for word categories other than nouns, such as adjectives, adverbs, etc.
2017 [68]	Gujarati	Unsupervised	Genetic Algorithm	A genetic algorithm is used.	-	-	-
2018 [60]	Gurumukhi	Supervised	K-NN	KNN classifier is used to find the similarity between vectors of input words and their meaning in Wordnet.	Punjabi Corpora of 100 sense tagged words is used.	The accuracy varies for each word, with the highest being 76.4% and the lowest being 53.6%.	The size of the dataset is too small.
2018 [61]	Punjabi	Supervised	Naive Bayes	Naive Bayes classifier, with Bow and collocation model as feature extraction technique, is used.	corpus of 150 ambiguous words having 6 or more senses taken from Punjabi word net	89% with BoW and 81% with the collocation model.	One word disambiguation per context.

Table 2. *Cont.*

Year (Ref.)	Language	Technique	Method	Specification	Dataset Used	Accuracy	Comments
2019 [69]	Hindi	Unsupervised	Word Embedding	Two-word embedding techniques, i.e., Skip-gram and CBow are used with cosine similarity to deduce the correct sense of the world.	-	52%	Semantic relations such as hypernyms, hyponyms, etc., are not used for the creation of sense vectors.
2019 [52]	Hindi	Knowledge-Based	Fuzzified Semantic Relations	Fuzzified semantic relations along with FHWN are used for WSD.	-	58–63%	There is uncertainty associated with fuzzy values. Values assigned to fuzzy memberships are based on the intuition of annotators.
2019 [53]	Hindi	Knowledge-Based	Lesk	Lesk algorithm is used to disambiguate the words.	A corpus of 3000 ambiguous sentences is used.	71.43%	POS tagger is not used
2019 [54]	Bengali	Knowledge-Based	Sense Induction	The semantic similarity measure is calculated for various sense clusters of ambiguous words.	A test set of 10 Bengali words is used.	63.71%	Classification of senses is not performed.
2021 [55]	Hindi	Knowledge-Based	Score-Based Modified Lesk	A scoring technique is utilized for advancing the performance of the Lesk algorithm.	-	-	Due to the segregation of only a part of the data from WordNet, the database needs to be queried repeatedly.
2021 [70]	Malyalam	Unsupervised	Semantic Features and Latent Dirichlet Allocation	An unsupervised LDA-based approach using semantic features has been applied for the target word sense disambiguation of the Malayalam language.	A dataset of 1147 contexts of polysemous words is used.	80%	LDA does not take into account the positional parameters within the context.

Table 2. *Cont.*

Year (Ref.)	Language	Technique	Method	Specification	Dataset Used	Accuracy	Comments
2021 [71]	Hindi	Unsupervised	Word Embeddings	Various word embedding technique has been used for WSD and experiments shows that Word2Vec performs better than all.	Hindi word embeddings were generated using articles sourced from Wikipedia.	54%	Further enhancements can be achieved by incorporating additional similarity metrics and incorporating sentence or phrase-level word embeddings into the approach.
2022 [67]	Hindi	Unsupervised	Co-operative Game Theory	Co-operative game theory along with Concept Net is used. It mitigated the influence of variations in membership values of fuzzy relations..	Health and tourism dataset and a manually created dataset from Hindi newspaper articles.	66%	-
2022 [56]	Hindi	Knowledge-Based		A complete framework named "HindiWSD" is developed in this that uses the knowledge-based modified Lesk algorithm.	A dataset of 20 ambiguous word along with Hindi WordNet is used.	71%	Dataset size is small.
2022 [72]	Hindi	Unsupervised	Genetic Algoritm	After pre-processing and creating the context bag and sense bag, GA is employed. In GA, selection, crossover and mutation are applied for the disambiguation of the word.	A manually created dataset is used.	80%	Only worked with nouns.

In the field of WSD for Hindi, the availability of high-quality data has been a challenge due to the resource-scarce nature of the language. However, there have been efforts to create and utilize datasets and benchmarks for Hindi WSD. Table 3 provide an overview about some common datasets and benchmarks that have been used or recognized in this field a:

Table 3. Data Sources available for Hindi WSD.

Data Source/Benchmark	Description
Hindi WordNet	Lexical database providing synsets and semantic relations for word senses in Hindi.
SemEval Hindi WSD Task	Part of the SemEval workshops, offering annotated datasets, evaluation metrics, and tasks for WSD in multiple languages.
Sense-Annotated Corpora	Manually annotated text segments where words are tagged with their corresponding senses from Hindi WordNet.
Cross-Lingual Resources	Leveraging resources from related languages with more data for WSD and transferring knowledge across languages.
Parallel Corpora	Using texts available in multiple languages to align senses and perform cross-lingual WSD.
Indigenous Corpora	Domain-specific or genre-specific corpora in Hindi, focusing on specific areas such as medicine, technology, or literature.
Supervised Approaches	Using a small annotated dataset for training models, often involving manually sense-tagged instances.
Unsupervised Approaches	Employing techniques such as clustering or distributional similarity without relying heavily on labeled data.
Contextual Embeddings	Utilizing pretrained models such as BERT to capture rich semantic information from large text corpora.

Because of the limitations in resources, the domain of Hindi WSD may not possess an equivalent abundance of universally accepted benchmarks as observed in more resource-endowed languages. As a result, researchers frequently modify techniques and methodologies drawn from other languages. Moreover, they occasionally amalgamate existing resources with data augmentation strategies to elevate their model's efficacy. The task of formulating more expansive and varied sense-annotated datasets and benchmarks continues to be a persisting challenge within this sphere.

4.4. Research Gaps and Future Scope

Hindi is a rich language in terms of users and information available in the Hindi language, and not much work has been performed on this. These are some of the research gaps, with the majority of the work involving nouns. Word lemmatization, which could improve accuracy even further, is not carried out, and one of the difficulties is understanding the idiomatic words. There is no standard sense annotated dataset available for supervised approaches. Using better methods or a hybrid model also has the potential to improve accuracy. Significant efforts have been dedicated to research and development for the English language, but Hindi, as the top fourth language in the world in terms of native speakers, is still in its infancy stage in the case of WSD. There is still a significant amount of work to be performed for the Hindi language. There is a lot of scope for improving accuracy, as well as other challenges, such as morphology, etc., that need to be solved.

5. Conclusions

This article summarizes several techniques utilized for the disambiguation of word senses based on Hindi literary sources. The classification of Hindi WSD tasks has categorized its methods into sections: supervised learning-based methods, knowledge-based methods, and unsupervised and supervised ones. Several types of knowledge-based, supervised, and unsupervised techniques are reviewed. Every approach has its own set of

rules for working and helps in solving a particular type of problem. In order to achieve superior outcomes with supervised methods, it is necessary to create an annotated dataset. Creating an annotated dataset can be both difficult and costly. However, the use of unannotated datasets with unsupervised approaches generally produces less favorable results than those produced using supervised techniques. Tackling resource-scarce languages effectively requires a knowledge-intensive approach. A comparative analysis of various approaches has been conducted, providing insights into the work undertaken by different researchers in the field. In conclusion, each category of WSD techniques offers distinct advantages and faces specific challenges. Supervised techniques excel in accuracy and fine-grained disambiguation but require labeled data and may struggle with generalization. Unsupervised techniques are flexible, scalable, and adapt well to languages with limited resources, yet they may encounter sense overlapping and lack semantic interpretation. Knowledge-based techniques leverage external resources effectively but heavily rely on the quality of knowledge bases. The choice of technique depends on task requirements, data availability, and language characteristics. Hybrid models, combining different techniques, can effectively address limitations and improve overall WSD performance, providing a tailored approach for specific applications and language contexts.

Author Contributions: Conceptualization, V.G. and N.M.; methodology, V.G.; software, R.K.; validation, V.G., S.P. and G.B.; formal analysis, N.M.; investigation, V.G.; resources, N.C.; data curation, V.G.; writing—original draft preparation, V.G. and S.P.; writing—review and editing, G.B.; visualization, R.K.; supervision, G.B.; project administration, N.C.; funding acquisition, N.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zipf, G.K. *Human Behavior and the Principle of Least Effort*; Addison-Wesley Press: Oxford, UK, 1949.
2. Wilks, Y.; Fass, D. The preference semantics family. *Comput. Math. Appl.* **1992**, *23*, 205–221. [CrossRef]
3. Navigli, R. Word sense disambiguation: A survey. *ACM Comput. Surv.* **2009**, *41*, 1459355. [CrossRef]
4. Vickrey, D.; Biewald, L.; Teyssier, M.; Koller, D. Word-sense disambiguation for machine translation. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), Vancouver, BC, Canada, 6 October 2005; pp. 771–778. [CrossRef]
5. Carpuat, M.; Wu, D. Improving statistical machine translation using word sense disambiguation. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007; pp. 61–72.
6. Pu, X.; Pappas, N.; Henderson, J.; Popescu-Belis, A. Integrating Weakly Supervised Word Sense Disambiguation into Neural Machine Translation. *Trans. Assoc. Comput. Linguist.* **2018**, *6*, 635–649. [CrossRef]
7. Plaza, L.; Jimeno-Yepes, A.J.; Díaz, A.; Aronson, A.R. Studying the correlation between different word sense disambiguation methods and summarization effectiveness in biomedical texts. *BMC Bioinform.* **2011**, *12*, 355. [CrossRef] [PubMed]
8. Madhuri, J.N.; Ganesh Kumar, R. Extractive Text Summarization Using Sentence Ranking. In Proceedings of the 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 1–2 March 2019; pp. 19–21. [CrossRef]
9. Carpineto, C.; Romano, G. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.* **2012**, *44*, 2071390. [CrossRef]
10. Sharma, N.; Niranjan, P.S. Applications of Word Sense Disambiguation: A Historical Perspective. *IJERT* **2015**, *3*, 1–4.
11. Sumanth, C.; Inkpen, D. How much does word sense disambiguation help in sentiment analysis of micropost data? In Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Lisboa, Portugal, 14 July 2015; pp. 115–121. [CrossRef]
12. Xu, G.; Yu, Z.; Yao, H.; Li, F.; Meng, Y.; Wu, X. Chinese Text Sentiment Analysis Based on Extended Sentiment Dictionary. *IEEE Access* **2019**, *7*, 43749–43762. [CrossRef]
13. Chifu, A.G.; Ionescu, R.T. Word sense disambiguation to improve precision for ambiguous queries. *Open Comput. Sci.* **2012**, *2*, 398–411. [CrossRef]
14. Asim, M.N.; Wasim, M.; Khan, M.U.G.; Mahmood, N.; Mahmood, W. The Use of Ontology in Retrieval: A Study on Textual, Multilingual, and Multimedia Retrieval. *IEEE Access* **2019**, *7*, 21662–21686. [CrossRef]

15. Advaith, V.; Shivkumar, A.; Sowmya Lakshmi, B.S. Parts of Speech Tagging for Kannada and Hindi Languages using ML and DL models. In Proceedings of the 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 8–10 July 2022. [CrossRef]
16. Gadde, S.P.K.; Yeleti, M.V. Improving statistical POS tagging using Linguistic feature for Hindi and Telugu Improving statistical POS tagging using linguistic features for Hindi and Telugu. In Proceedings of the ICON-2008: International Conference on Natural Language Processing, Pune, India, 20–22 December 2008.
17. Banerjee, S.; Pedersen, T. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In Proceedings of the Computational Linguistics and Intelligent Text Processing, Mexico City, Mexico, 17–23 February 2002; Gelbukh, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 136–145.
18. Lesk, M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the 5th Annual International Conference on Systems Documentation, Toronto, ON, Canada, 1 June 1986; pp. 24–26. [CrossRef]
19. Mittal, K.; Jain, A. Word Sense Disambiguation Method Using Semantic Similarity Measures and Owa Operator. *ICTACT J. Soft Comput.* **2015**, *5*, 896–904. [CrossRef]
20. Mccarthy, D.; Carroll, J. Adjectives Using Automatically Acquired Selectional Preferences. *Comput. Linguist.* **2003**, *29*, 639–654. [CrossRef]
21. Ye, P.; Baldwin, T. Verb Sense Disambiguation Using Selectional Preferences Extracted with a State-of-the-art Semantic Role Labeler. In Proceedings of the Australasian Language Technology Workshop 2006, Sydney, Australia, 4–6 December 2006; pp. 139–148.
22. Sarika; Sharma, D.K. A comparative analysis of Hindi word sense disambiguation and its approaches. In Proceedings of the International Conference on Computing, Communication & Automation, Pune, India, 26–27 February 2015; pp. 314–321.
23. Walker, J.Q., II. A node-positioning algorithm for general trees. *Softw. Pract. Exp.* **1990**, *20*, 685–705. [CrossRef]
24. Parameswarappa, S.; Narayana, V.N. Decision List Preliminaries of the Kannada Language and the Basic. 2013, Volume 2. Available online: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7620a95796c2eae4a94498fa779b00e2b25c957a (accessed on 21 May 2023).
25. Yarowsky, D. Hierarchical decision lists for word sense disambiguation. *Lang. Resour. Eval.* **2000**, *34*, 179–186.
26. Singh, R.L.; Ghosh, K.; Nongmeikapam, K.; Bandyopadhyay, S. A Decision Tree Based Word Sense Disambiguation System in Manipuri Language. *Adv. Comput. Int. J.* **2014**, *5*, 17–22. [CrossRef]
27. Rawat, S.; Kalambe, K.; Kawade, G.; Korde, N. Supervised word sense disambiguation using decision tree. *Int. J. Recent Technol. Eng.* **2019**, *8*, 4043–4047. [CrossRef]
28. Thwet, N.; Soe, K.M.; Thein, N.L. System Using Naïve Bayesian Algorithm for Myanmar Language. *Int. J. Sci. Eng. Res.* **2011**, *2*, 1–7.
29. Le, C.A.; Shimazu, A. High WSD accuracy using Naive Bayesian classifier with rich features. In Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation PACLIC 2004, Tokyo, Japan, 8–10 December 2004; pp. 105–113.
30. Popov, A. Neural network models for word sense disambiguation: An overview. *Cybern. Inf. Technol.* **2018**, *18*, 139–151. [CrossRef]
31. Kumar, S.; Kumar, R. Word Sense Disambiguation in the Hindi Language: Neural Network Approach. *Int. J. Tech. Res. Sci.* **2021**, *1*, 72–76. [CrossRef]
32. Kumar, M.; Sankaravelayuthan, R.; Kp, S. Tamil word sense disambiguation using support vector machines with rich features. *Int. J. Appl. Eng. Res.* **2014**, *9*, 7609–7620.
33. Decadt, B.; Hoste, V.; Daelemans, W.; van den Bosch, A. GAMBL, genetic algorithm optimization of memory-based WSD. In Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, 25–26 July 2004; pp. 108–112.
34. Fix, E.; Hodges, J.L. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *Int. Stat. Rev. Rev. Int. Stat.* **1989**, *57*, 238. [CrossRef]
35. Hamming, R.W. Error detecting and error correcting codes. *Bell Syst. Tech. J.* **1950**, *29*, 147–160. [CrossRef]
36. Revesz, P.Z. A Generalization of the Chomsky-Halle Phonetic Representation using Real Numbers for Robust Speech Recognition in Noisy Environments. In Proceedings of the 27th International Database Engineered Applications Symposium, Heraklion, Greece, 5–7 May 2023; pp. 156–160. [CrossRef]
37. Brody, S.; Navigli, R.; Lapata, M. Ensemble methods for unsupervised WSD. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 17–18 July 2006; Volume 1, pp. 97–104. [CrossRef]
38. Freund, Y.; Schapire, R.E. A Short Introduction to Boosting. 1999. Available online: https://api.semanticscholar.org/CorpusID:9621074 (accessed on 20 December 2022).
39. Martín-Wanton, T.; Berlanga-Llavori, R. A clustering-based approach for unsupervised word sense disambiguation. *Proces. Leng. Nat.* **2012**, *49*, 49–56.
40. Lin, D. Automatic retrieval and clustering of similar words. *Proc. Annu. Meet. Assoc. Comput. Linguist.* **1998**, *2*, 768–774. [CrossRef]
41. Pantel, P.A. Clustering by Committee. 2003, pp. 1–137. Available online: https://www.patrickpantel.com/download/papers/2003/cbc.pdf (accessed on 25 January 2023).

42. Silberer, C.; Ponzetto, S.P. UHD: Cross-lingual word sense disambiguation using multilingual Co-occurrence graphs. In Proceedings of the 5th International Workshop on Semantic Evaluation, Uppsala, Sweden, 12 July 2010; pp. 134–137.
43. Miller, G.A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
44. Bhattacharyya, P. IndoWordnet. In *The WordNet in Indian Languages*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 3785–3792. [CrossRef]
45. Sinha, M.; Reddy, M.K.; Bhattacharyya, P.; Pandey, P.; Kashyap, L. Hindi Word Sense Disambiguation. 2004. Available online: https://api.semanticscholar.org/CorpusID:9438332 (accessed on 25 January 2023).
46. Singh, S.; Siddiqui, T.J. Evaluating effect of context window size, stemming and stop word removal on Hindi word sense disambiguation. In Proceedings of the 2012 International Conference on Information Retrieval & Knowledge Management, Kuala Lumpur, Malaysia, 13–15 March 2012; pp. 1–5. [CrossRef]
47. Kumar Vishwakarma, S.; Vishwakarma, C.K. A Graph Based Approach to Word Sense Disambiguation for Hindi Language. *Int. J. Sci. Res. Eng. Technol.* **2012**, *1*, 313–318. Available online: www.ijsret.org (accessed on 25 January 2023).
48. Singh, S.; Singh, V.K.; Siddiqui, T.J. *Hindi Word Sense Disambiguation Using Semantic Relatedness Measure BT-Multi-Disciplinary Trends in Artificial Intelligence*; Ramanna, S., Lingras, P., Sombattheera, C., Krishna, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 247–256.
49. Singh, S.; Siddiqui, T.J. Role of semantic relations in Hindi Word Sense Disambiguation. *Procedia Comput. Sci.* **2015**, *46*, 240–248. [CrossRef]
50. Sawhney, R.; Kaur, A. A modified technique for Word Sense Disambiguation using Lesk algorithm in Hindi language. In Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi, India, 24–27 September 2014; pp. 2745–2749. [CrossRef]
51. Gautam, C.B.S.; Sharma, D.K. Hindi word sense disambiguation using lesk approach on bigram and trigram words. In Proceedings of the International Conference on Advances in Information Communication Technology & Computing, Bikaner, India, 12–13 August 2016; pp. 1–5. [CrossRef]
52. Jain, G.; Lobiyal, D.K. Word sense disambiguation of Hindi text using fuzzified semantic relations and fuzzy Hindi WordNet. In Proceedings of the 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 10–11 January 2019; pp. 494–497. [CrossRef]
53. Sharma, P.; Joshi, N. Knowledge-Based Method for Word Sense Disambiguation by Using Hindi WordNet. *Eng. Technol. Appl. Sci. Res.* **2019**, *9*, 3985–3989. [CrossRef]
54. Sau, A.; Amin, T.A.; Barman, N.; Pal, A.R. Word sense disambiguation in bengali using sense induction. In Proceedings of the 2019 International Conference on Applied Machine Learning (ICAML), Bhubaneswar, India, 25–26 May 2019; pp. 170–174. [CrossRef]
55. Tripathi, P.; Mukherjee, P.; Hendre, M.; Godse, M.; Chakraborty, B. Word Sense Disambiguation in Hindi Language Using Score Based Modified Lesk Algorithm. *Int. J. Comput. Digit. Syst.* **2021**, *10*, 939–954. [CrossRef]
56. Yusuf, M.; Surana, P.; Sharma, C. HindiWSD: A Package for Word Sense Disambiguation in Hinglish & Hindi. In Proceedings of the 6th Workshop on Indian Language Data: Resources and Evaluation (WILDRE-6), Marseille, France, 20–25 June 2022; pp. 18–23.
57. Purohit, A.; Yogi, K.K. A Comparative Study of Existing Knowledge Based Techniques for Word Sense Disambiguation. In Proceedings of the International Joint Conference on Advances in Computational Intelligence, Online, 23–24 October 2021; Uddin, M.S., Jamwal, P.K., Bansal, J.C., Eds.; Springer Nature: Singapore, 2022; pp. 167–182.
58. Singh, S.; Slddiqui, T.J.; Sharma, S.K. Naïve bayes classifier for hindi word sense disambiguation. In Proceedings of the 7th ACM India Computing Conference, Nagpur, India, 9 October 2014. [CrossRef]
59. Sarika; Sharma, D.K. Hindi word sense disambiguation using cosine similarity. In Proceedings of the Advances in Intelligent Systems and Computing, Athens, Greece, 29–31 August 2016.
60. Walia, H.; Rana, A.; Kansal, V. A Supervised Approach on Gurmukhi Word Sense Disambiguation Using K-NN Method. In Proceedings of the 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 11–12 January 2018; pp. 743–746. [CrossRef]
61. pal Singh, V.; Kumar, P. Naive Bayes classifier for word sense disambiguation of Punjabi Language. *Malaysian J. Comput. Sci.* **2018**, *31*, 188–199. [CrossRef]
62. Mishra, B.K.; Jain, S. *Word Sense Disambiguation for Hindi Language Using Neural Network BT-Advancements in Smart Computing and Information Security*; Rajagopal, S., Faruki, P., Popat, K., Eds.; Springer Nature: Cham, Switzerland, 2022; pp. 14–25.
63. Mishra, N.; Yadav, S.; Siddiqui, T.J. An Unsupervised Approach to Hindi Word Sense Disambiguation. In Proceedings of the First International Conference on Intelligent Human Computer Interaction, Rome, Italy, 20–23 January 2009.
64. Jain, A.; Lobiyal, D.K. Unsupervised Hindi word sense disambiguation based on network agglomeration. In Proceedings of the 2015 International Conference on Computing for Sustainable Global Development, INDIACom 2015, New Delhi, India, 11–13 March 2015.
65. Nandanwar, L. Graph connectivity for unsupervised Word Sense Disambiguation for Hindi language. In Proceedings of the ICIIECS 2015—2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems, Coimbatore, India, 19–20 March 2015.

66. Jain, A.; Lobiyal, D.K. Fuzzy Hindi wordnet and word sense disambiguation using fuzzy graph connectivity measures. *ACM Trans. Asian Low-Resource Lang. Inf. Process.* **2015**, *15*, 2790079. [CrossRef]
67. Jain, G.; Lobiyal, D.K. Word Sense Disambiguation Using Cooperative Game Theory and Fuzzy Hindi WordNet Based on ConceptNet. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2022**, *21*, 3502739. [CrossRef]
68. Vaishnav, Z.B. Gujarati Word Sense Disambiguation Using Genetic Algorithm. 2017. Available online: https://api.semanticscholar.org/CorpusID:212514785 (accessed on 25 January 2023).
69. Kumari, A.; Lobiyal, D.K. Word2vec's Distributed Word Representation for Hindi Word Sense Disambiguation. In Proceedings of the 16th International Conference, ICDCIT 2020, Bhubaneswar, India, 9–12 January 2020. [CrossRef]
70. Sruthi, S.; Kannan, B.; Paul, B. Improved Word Sense Determination in Malayalam using Latent Dirichlet Allocation and Semantic Features. *ACM Trans. Asian Low-Resource Lang. Inf. Process.* **2022**, *21*, 3476978. [CrossRef]
71. Kumari, A.; Lobiyal, D.K. Efficient estimation of Hindi WSD with distributed word representation in vector space. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6092–6103. [CrossRef]
72. Bhatia, S.; Kumar, A.; Khan, M. Role of Genetic Algorithm in Optimization of Hindi Word Sense Disambiguation. *IEEE Access* **2022**, *10*, 3190406. [CrossRef]
73. Jha, P.; Agarwal, S.; Abbas, A.; Siddiqui, T. Comparative Analysis of Path-based Similarity Measures for Word Sense Disambiguation. In Proceedings of the 2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada, India, 18–20 March 2023; pp. 1–5. [CrossRef]