UtilitasMathematica

ISSN 0315-3681 Volume 122, 2025

WORD SENSE DISAMBIGUATION: A ROADMAP

SRUTHI S.

Department of Computer Applications,
Cochin University of Science and Technology
Kochi, India

B. KANNAN

Department of Computer Applications,
Cochin University of Science and Technology
Kochi, India

BINU PAUL

School of Engineering
Cochin University of Science and Technology
Kochi, India

Abstract

Words that can be used with multiple senses are commonly referred to as polysemic words. Word Sense Disambiguation (WSD) is the process of finding the intended sense of a polysemic word in a given context. It is one of the most challenging problems in the area of natural language processing since it is an intermediate step in Machine Translation, Speech and Text processing, Information Retrieval and Extraction, etc. The research in this field has been ongoing for more than 60 years, and many changes can be seen in the techniques adopted. These changes are driven by the domain in which the WSD is applied and the availability of knowledge sources like corpora and machine-readable dictionaries. This paper aims to give a global perspective on the significant state-of-the-art techniques on Word Sense Disambiguation.

***Keywords*:** Word Sense Disambiguation, Polysemic Word, Indian Language Computing, Review, Natural Language Processing.

1. Introduction

In general, words can have multiple meanings depending upon the context. Word sense disambiguation is the process of resolving ambiguity in the meaning of a word. WSD is one of the essential elementary tasks in lexical analysis. This is an intermediate step that contributes to many computational linguistic problems rather than a stand-alone problem. An improved WSD will be helpful to many NLP problems like Machine Translation, Information Extraction, and information retrieval.

1.1. *WSD in Machine translation*

Work in WSD as an intermediate task in Machine Translation started in the early 1940s. Machine translation is a field in computational linguistics which converts text/speech in one language to another. The use of WSD for machine translation has been studied by [(Carpuat 2005)] and [(Chan et al., 2007)].

1.2. *WSD in Information retrieval and information extraction*

The query words in IR will have multiple senses, which will affect the retrieval precision. Also, the words in the query will have close meaning to other words which are not in the query. WSD can resolve these problems by correctly identifying these issues and thus improves retrieval accuracy (Agrawal & Srikant, 1994) [4]. Some of the IR-related applications are cross-language IR [(Ide, 2000)], [(Ide et al., 2002)], [(Bhattacharya et al., 2004.)], Question answering systems [(Pasca & Harabagiu., 2001)], Document classification [(Bloehdorn & Hotho, 2004)], etc.

1.3. *WSD in speech and text processing*

Disambiguation of words is very useful in phonetization of words [(Seneff, 1992)], spelling and grammatical error corrections [(Chodorow & Leacock, 2000)], sentence case change, etc.

An extensive survey in this area was conducted by Roberto Navigli in 2009 [(Navigli, 2009)]. Our paper reviews the progress of various disambiguation methods and surveys major Word Sense Disambiguation The article is structured as follows: Section 2 deals with the problem overview. Sections 3 and 4 cover prominent approaches used in Word Sense Disambiguation and works conducted in Indian languages. Sections 5 and 6 describe open problems in WSD and the findings of the paper, respectively. Section 7 contains the conclusion and future scope.

2. Problem Overview

WSD is an AI problem that can be treated as an intermediate step in NLP applications like Machine Translation, Information retrieval, etc., or a stand-alone problem. Two variations in WSD are All-words disambiguation and target word disambiguation. In All-words disambiguation, all the words within the context are disambiguated, whereas the target word disambiguation, only the targeted word. Disambiguation is carried out based on contextual information. The selection and performance of the disambiguation technique heavily depend on the available knowledge sources and the application for which WSD is conducted. As the quality and quantity of knowledge sources improve, the chance for a better WSD result also gets improved. Generally, supervised techniques are used in target word disambiguation while unsupervised and knowledge-based techniques in all-words disambiguation. The main steps involved in the pre-processing phase of WSD are stop word removal, tokenization, stemming, POS tagging, etc. The following section covers various approaches to WSD.

3. Approaches to Word Sense Disambiguation

Various approaches to WSD that have been widely used are discussed in this section. These techniques are classified based on the type of knowledge sources or resources used for disambiguation. The general hierarchy is given in Fig.1.

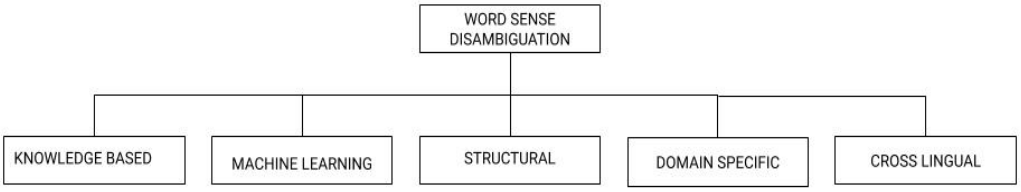


Fig. 1. Different approaches to Word Sense Disambiguation

3.1. *Knowledge-based WSD*

Knowledge-based methods use different knowledge sources such as machine-readable dictionaries or sense inventories, Wikipedia, thesauri, WordNet, ontologies, etc. Various models on knowledge-based WSD are shown in Fig.2.

3.1.1. Lesk Algorithm/ Overlap-based methods: The Lesk algorithm [(Lesk, 1986, June)] is an overlap-based method that evolved in 1986. In this algorithm, the overlap between senses of the target word from the knowledge source and two-word contexts from the input sentence are compared. The sense definition, which has the highest overlap with the context words, is assumed to be the correct sense. The disadvantage with this technique is that it is very much dependent on the definitions of the senses in the dictionary. The absence or presence of a word in the sense definition could affect the results primarily. This method achieved 50-70% of accuracy. Incorporating Wordnet information leads to improved accuracy for WSD [(Banerjee & Pedersen, 2002.)]. Since dictionary definitions are very short and precise, the overlap-based methods will not work effectively. In order to resolve this issue, the Wordnet information is also adapted to connect to more words. The definitions of synonym, hypernym, hyponym, holonym, meronym, troponym, and attribute relations of the ambiguous words were also considered with this technique. This algorithm outperforms the Lesk algorithm in disambiguating nouns in the Senseval-2 task.

Some variants of the Lesk algorithm were also proposed in [(Patwardhan et al., 2003)], [(Zouaghi et al., 2012)], etc. In their study, Jimenez et al developed a knowledge-based WSD. They found that WSD performance can be improved by changing the window selection procedure, extending sense definitions with co-occurring words, or disambiguating only domain words. In [(Basile et al., 2014)] the authors adopted Babel-Net, an extensive multilingual semantic network built exploiting WordNet and Wikipedia. Bable-Net also incorporates the encyclopedic concepts from Wikipedia. Based on the SemEval-2013 evaluation, this model outperformed the most frequent sense baseline and the simplified version of the Lesk algorithm for Multilingual Word Sense Disambiguation.

Some of the latest works in WSD using Lesk algorithms include [(Singh et al., 2021.)], and [(Poluru et al., 2021.)].

TABLE I. KNOWLEDGE-BASED WSD WORKS

Authors and year	**Techniques and features used**	**Performance or results**
Lesk,1986	Overlap based method known as Lesk algorithm, Considering the overlap between gloss definitions and context of ambiguous words, gloss definitions are taken from Webster's 7th collegiate, Collins English Dictionary,	Accuracy 50 - 70 %

	Oxford's Advanced Learner's Dictionary of Current English.	
Banerjee,2002	Adapted Lesk Algorithm using Wordnet Relations, Outperforms Lesk algorithm in disambiguating nouns in Senseval2 English task.	Accuracy 83% improved over Lesk algorithm
Jimenez, 2014	Improved WSD by changing the window selection, co-occuring words, domain words etc. Also introduced a novel optimization algorithm that takes less time than the Lesk algorithm.	Senseval2-P 50.2, R-47.9, F1-49.0 Senseval3- P-39.4, R-37.5, F1-38.4
Basile, 2014	Enhanced Lesk algorithm which incorporates BableNet, which has both linguistic and encyclopedic knowledge.	Improvement over Lesk algorithm in SemEval-2013 multilingual WSD (English and Italian) Task.

3.1.2. Selectional preferences: They are restrictions on the possible relations between word categories. For example, Eat-Food, Drink-Liquid, Assassinate-Person are some of the selectional restrictions. Such relations will be helpful to find out the false sense of words. These restrictions can be easily learned by finding the word-to-word relations. This can be done using word-to-word frequency count or conditional probability. This knowledge can be generalized to word-to-class or class-to-class problems. Selectional restriction approaches do not work as well as the Lesk Algorithm for WSD [(Resnik, 1997)], [(Carroll & McCarthy, 2000)] et al. from their study on Senseval data, found out that selectional preferences are not a stand-alone solution for WSD. A WSD system requires other knowledge sources also.

Yoonjung Choi [(Choi et al., 2017)] et al. developed a coarse-grained WSD technique for sentiment analysis. The system aims to capture the +/- effect of a word in different contexts. This work seeks to find the chances of a verb to co-occur with certain types of arguments, which decide the positive or negative (+/-) sense of the verb. This was done with the help of selectional preferences, modeled using Latent Dirichlet Allocation (LDA).

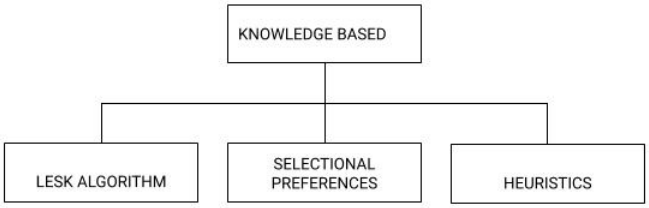


Fig. 2. Different knowledge-based approaches to Word Sense Disambiguation

3.1.3. Heuristics for Word Sense Disambiguation: Most-frequent sense is one of the most used heuristics in many WSD systems. It is based on the assumption that a word will have one meaning which occurs frequently. But (Ng, 1997) et al. predicted that with an accurate sense-tagged corpus, one could outperform these most-frequent sense classifiers, as they may not be correct in all cases. According to Gale (Gale et al., 1992), there is a strong probability for an ambiguous term having the same meaning across the same discourse. This is known as one sense per discourse, a heuristic that was introduced in 1992. This was based on the assumption that a polysemous word will have exactly one meaning under certain definitions of collocations.

3.2. *Machine Learning based WSD*

Machine Learning (ML) algorithms do not use knowledge-based resources such as Machine-Readable Dictionaries, ontologies, WordNet, etc. These techniques are based on the assumption that word sense disambiguation can be viewed as a classification task, where the words should be classified into different classes called senses [(Navigli, 2009)]. The Fig.3**.** shows major works under Machine Learning based WSD.

***3.1.1. Supervised WSD*:** In supervised methods, ML algorithms are used to build a classifier with the help of a set of labeled data, which aims to classify the senses correctly.

Naive Bayes: This is a probabilistic classifier based on Bayes Theorem. Naive Bayes classifier works on the assumption that the sense of the ambiguous word, which gives maximum conditional probability along with the context words, will be the most appropriate sense for that context.

Suppose X = (x_1; x_2; …; x_n) is a feature vector, with n independent variables and y, the class variable. Then, Bayes theorem states the following relationship as in equation 1.

$$P(y \mid (x_1, x_2, ..., x_n)) = \frac{P(x_1, x_2, ..., x_n \mid y)\, P(y)}{P(x_1, x_2, ..., x_n)} \quad (1)$$

From this equation, a classifier model can be derived mathematically as in equation (2).

$$\hat{y} = \arg\max P(y) \prod_{i=1}^{n} P(X_i \mid y) \quad (2)$$

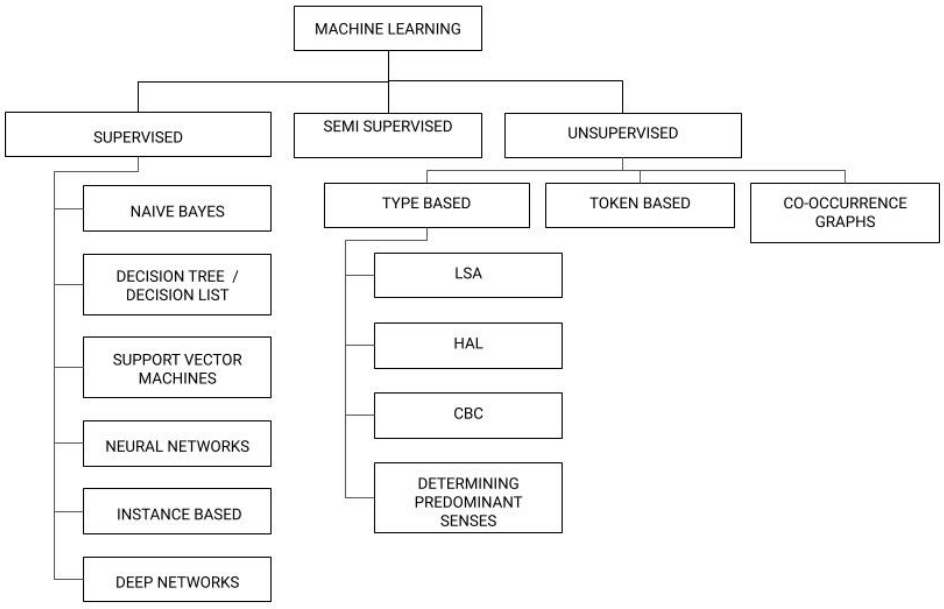


Fig. 3. Machine Learning based WSD

Gale et al. developed the first WSD system based on the Naive Bayes approach (Gale et al., 1992), where an Information Retrieval approach is used for WSD. Here, instead of documents, the context 'c' of an ambiguous word was sorted by the following score.

$$score(c) = \prod_{token\ in\ c} \frac{Pr(token|sense_1)}{Pr(token|sense_2)}$$

Ted Pederson, in 2000, proposed an ensemble of Naive Bayesian classifiers and showed better results than the state-of-the-art (Pedersen, 2000). According to the authors, the probability of observing a certain combination of contextual features with a particular sense can be expressed as the following equation, where F_i indicates the features and S, the sense:

$$p(F_1, F_2, \ldots, F_n, S) = p(S) \prod_{i=1}^{n} p(F_i|S)$$

He combined nine ensembles of Naive Bayesian classifiers, each with different left and right contextual window sizes. The test was conducted on the 'Line' and 'Interest' data set, and an accuracy of 89% and 88% were obtained.

Gerard Escudero compared Naive Bayes and exemplar-based approaches to WSD in 2000 (Escudero et al., 2000). They used the k-NN algorithm with hamming distance in the exemplar approach. From their study, exemplar algorithms were found to perform better than Naive Bayesian classifiers when additional features and example weighting schemes are added. Many researchers adopted NB classifiers for WSD tasks in various languages. Some of them are [(Le & Shimazu, 2004)], [(Aung et al., 2011)], and [(Elmougy et al., 2008.)].

Decision trees and Decision Lists: Decision Trees are predictive models developed by Quinlan in 1986(Quinlan, 1986). A decision tree predicts the classification rules in a tree structure that recursively partitions the training data set. The most famous decision tree algorithms are ID3 and C4.5, which use attribute selection measures such as Information gain and Gain ratio. In 2001, Ted Pederson (Pedersen, 2001) utilized a decision tree classifier using bigrams with collocation as features. They have also concluded that informative feature sets contribute a lot to the WSD process rather than selecting an ML algorithm.

Decision Lists consist of rules for classifying word senses [(Rivest, 1987)]. These will be weighted using if-then-else rules. A training set is used to induce the set of features. The ambiguous word can be represented as a vector. Then the list is checked, and the feature with the highest value is taken as the correct sense of the word. Yarowsky et al. (Yarowsky, 1994) proposed this method for accent restoration in Spanish and French lexical ambiguity resolution. The hierarchical decision list approach can also be used for WSD (Yarowsky, 2000), where conditional branching is used. This algorithm was effectively utilized for the Senseval dataset. Using this approach, a precision of 0.7 is obtained with the DSO (Agirre & Martinez, 2000) corpus.

Support vector machines (SVM): In 1992, Boser et al. [(Boser et al., 1992.)] developed a classifier that maximizes the margin between the training patterns and the decision boundary, known as SVM. SVMs are linear classifiers, which can be used with relatively

large features without too much computation.SVM works on the underlying assumption that maximizing the margin will minimize the loss. This binary classifier uses a hyperplane located in the hyperspace where the difference between positive and negative examples is maximum. The points which lie on the maximum margins are known as support vectors. SVM considers the very extreme cases, i.e., support vectors that lie very close to the decision boundary, which gives them a chance for better performance.WSD can be treated as a multi-class problem since a word can have more than one meaning. Hence the problem is broken down into several binary-class problems. We use Kernelized SVM for non-linearly separable data. The concept is that if we have some non-linearly separable data in one dimension, we can transform this data into two dimensions, and the data will become linearly separable in two dimensions. This is done by mapping each one-dimensional data point to a corresponding two-dimensional ordered pair. This can be extended to multiple dimensions. This is one of the best examples of kernel-based methods.

Yoong Keok Lee and Hwee Tou Ng [(Lee & Ng, 2002)], in 2002, conducted a study on supervised WSD techniques. They had concluded that the performance of algorithms varies as the feature selection criteria changes. Support vector machines performed better with collocation information, while Naive Bayesian classifiers worked well with feature selection. In Senseval-3, authors Yoong Keok Lee et al. [(Lee et al., 2004.)] developed a Support Vector Machine with the aid of multiple knowledge sources. The multiple knowledge sources used were parts-of-speech of the adjacent words, within a distance of 3 from the right and left to the target word, unigrams surrounding the micro context, collocations within the micro context, and parse tree generated by the Charniak statistical parser. They found that SVM performs much better with multiple knowledge sources. Mahesh Joshi et al. [(Joshi et al., 2005)] compared 5 ML techniques with SVM, namely NB, C4.5 Decision Tree, Decision Lists, and Boosting algorithms applied to Medical Domain. Features used are unigrams and bigrams selected using different frequency cut-off values and window size. SVM was the most accurate among other models. An SVM implementation for WSD has been performed by [(Zhong & Wang., 2020)] using multiple kernel learning (MKL) techniques. They used Bag-of-words, Sequence, and Tree kernel approaches for WSD. The authors experimented their work on Senseval / Semval datasets.

Neural Networks: Neural networks [(Cottrell, 1985.)] follow a connectionist approach in which they contain a large number of interconnected neurons. The pairs of feature values and expected responses will be the input to the network. The aim is to partition the test cases into non-overlapping sets based on the training contexts. This is done by changing the weights corresponding to each input feature and obtaining a high activation value for the desired output. A perceptron is the most straightforward kind of artificial neural network. A multilayer neural network/ multilayer perceptron consists of at least three layers, and each internal node consists of a non-linear activation function. They use the back-propagation technique for training [(Rumelhart et al., 1986)]. The input to unit j, Oj, is the product of the weights and the non-linear activation function of the outputs yi that are connected to j, as shown in (3)

$$o_j = \sum y_i w_{ji} \qquad (3)$$

Each unit will be having a real-valued output, which is an activation function of its total input, given in (4).

$$y_j = \frac{1}{1 + e^{-x_j}} \qquad (4)$$

Neural networks aim to find a set of weights for each observation, that the output produced should be the same as that of the desired output. As the total error can be found out only for the output layer, the backward pass propagates the error derivatives from the output layer back to the bottom layers. In WSD, the text data or the feature values and expected labels/senses will be the input to the network. The aim is to partition the test cases into non-overlapping sets based on the training contexts. This is done by changing the weights corresponding to each input feature and obtaining a high activation value for the expected output.

Veronis and Ide, in 1990, built a huge neural network model extracted from machine-readable dictionaries and used them for WSD [(Veronis & Ide, 1990)]. Some of the significant drawbacks of neural networks are setting the threshold value, a large quantity of data, difficulty in interpreting results, etc.

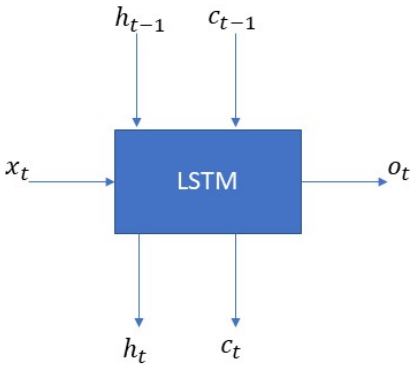
Bengio et al. [(Bengio et al., 2003)] developed a neural language probabilistic model, which helps in reducing the curse of dimensionality. Each word in the vocabulary is represented as a feature vector and the joint probability of word sequences in terms of feature vectors.

TABLE II. MACHINE LEARNING-BASED WSD WORKS

Authors and year	**Techniques and features used**	**Performance or results**
Gale et al.1992	Naive Bayes technique with context window size increased to 50 words and varying training sample size	Accuracy improved from 86% to 90% when context window size increased to 50 words, As the number of training samples increased, accuracy increased from 86% to 90%
Ted Pederson 2000	9 Ensembles of Naive Bayes classifier, 'line' and 'interest' dataset was used, 5-fold-cross validation is performed with features varying from left and right bag-of-words	88% accuracy on 'line' test data, 89% accuracy on 'interest' test data
Gerard Escudero 2000	Naive Bayes, Exemplars based on k-NN with hamming distance to measure closeness	Exemplar based method outperformed NB
(Le & Shimazu, 2004)	NB Algorithm with features as ordered words in local context and collocations etc.	92.3% accuracy (on Interest, line, hard, serve corpus), 66.4% accuracy for verbs and 72.7% accuracy for nouns (DSO corpus)
Yoong Koek Lee 2004	SVM with multiple features like POS tags, single surrounding words, local	For English tasks (Precision and Recall), scores of 0.724 and 0.788 were obtained for fine-grained and

	collocations, and syntactic relations obtained from Charniak's parser., Senseval-3 English lexical sample task, Multilingual Task.	coarse-grained, respectively. For the multilingual lexical sample task, recall (and precision) of 0.634 for the translation subtask and 0.673 for the translation and sense subtask were obtained.
Mahesh Joshi 2005	Comparison between SVM and 5 other ML (NB, C4.5 Decision Tree, Decision Lists, and Boosting) algorithms applied to Medical Domain. Features used are unigrams and bigrams selected using different frequency cut-off values and window size.	The SVM model outperformed other classifiers.

Deep networks and word embeddings: One of the limitations with multilayer perceptrons is that there is no memory associated with these models. This limitation raises problems in the processing of sequential data. Recurrent Neural Networks addresses that issue by including a feedback loop that serves as a kind of memory. LSTMs are a special kind of recurrent neural network with short-term and long-term memory components. A single LSTM unit is called a cell, which consists of an input gate, an output gate, and a forget gate, as shown in Figure. The cell remembers values over arbitrary time intervals and the three gates are responsible for regulating the flow of information through the cells. The LSTM network performs the same task for each element in a sequence whose output depends on the input and the previous state of the memory. Several techniques for WSD based on deep neural networks have been evolved in recent years. Most of them depend on word embedding concepts.



$$f_t = \sigma_g\,(W_f \times x_t + U_f \times h_{t-1} + b_f)$$
$$i_t = \sigma_g\,(W_i \times x_t + U_i \times h_{t-1} + b_i)$$
$$o_t = \sigma_g\,(W_o \times x_t + U_o \times h_{t-1} + b_o)$$
$$c'_t = \sigma_c\,(W_c \times x_t + U_c \times h_{t-1} + b_c)$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t$$
$$h_t = o_t \cdot \sigma_c(c_t)$$

f_t *is the forget gate*
i_t *is the input gate*
o_t *is the output gate*
c_t *is the cell state*
h_t *is the hidden state*

Fig. 4. LSTM input outputs and the corresponding equations for a single timestep [101].

Some of the major word embedding techniques evolved are

Word2vec: This was a breakthrough in NLP, came in 2013, by Mikolov et al. [(Mikolov et al., 2013)], a technique for the distributed representation of words in vector space. In Word2vec, each word in the dataset will be converted into a list of numbers called vectors which were generated with the help of a neural network model. In this model, each word vector is generated by measuring the cosine similarity between words. The 2 main variants of Word2vec are the skip-gram model and the continuous-bag-of-words model. The skip-gram model tries to predict the context of the current word. The CBOW model predicts the present word using the context.

Glove: In this model, the corpus's statistics of the word co-occurrences and how frequently they appear in particular contexts are studied and utilized by assigning a contextual window [(Pennington et al., 2014)]. Those words which are far from the window will be given lesser weights. Later matrix factorization is applied in this word-context matrix.

ELMo(Embeddings from Language Models): They use the two-layer bi-directional language models and character convolutions for generating word representations [(Peters et al., 2018)]. They are functions of the entire input sequence. With the help of ELMo, the generated word embeddings can be utilized for various NLP task-specific applications, which will result in significant improvement in the state-of-the-art results.

In 2017, Xue-Ren SUN et al. [(Sun et al.,)] developed an LSTM (Long Short-term memory) network-based WSD Model. Their paper replaced the ambiguous word with its synonyms and classified it as right or wrong. For this paper, they have used CBOW (Continuous Bag-of-Words) Word2vec model as the first layer. These vector representations are then fed to an LSTM model followed by an MLP model whose output is used for document classification.
It has been shown that Bidirectional LSTM can be effectively used in WSD [(Kågebäck & Salomonsson, 2016)] with word embeddings obtained from Glove (Global vectors for Word Representation) in which each word will correspond to a real-valued vector.
Alessandro Raganato et al. analyzed WSD as a neural sequence labeling problem [(Raganato et al., 2017.)]. They have found that instead of treating WSD for each word separately, a sequence learning approach can be used to disambiguate all ambiguous words jointly after training. They have experimented with a Bidirectional LSTM tagger, an attentive bidirectional LSTM, and encoder and decoder models.
Gloss knowledge and contextual knowledge have been efficiently used using a co-attention mechanism in [(Luo et al., 2018.)]. The authors have also extended this into a hierarchical architecture and included word level and sentence level information which have significant importance in WSD.Some notable works in deep learning based WSD are enlisted in table III.

TABLE III. DEEP LEARNING-BASED WSD WORKS

Authors and year	**Techniques and features used**	**Performance or results**
Alessandro Raganto 2017	Neural sequence models like attentive bidirectional LSTM taggers, encoder-decoder models, etc. Features used were	These models outperformed traditional supervised models.

	POS tags, coarse-grained semantic labels, etc. Test sets- SE2, SE3, SE7, SE13, SE15	
Xue-Ren SUN	Chinese WSD using BiLSTM	0.78 accuracy obtained with LSTM, embedding size=256, batch_size=100, with a drop-out rate of 0.5.
Fuli Luo et al.	Gloss knowledge with hierarchical co-attention (context to gloss, gloss to context, word-level, and sentence level)	Achieved State of the art results on All-words English WSD.
Mikael Kageback 2016	BiLSTM + Glove embeddings	Achieved State of the art results.
Sawankumar 2019 [47]	Zero-shot learning using a supervised classification model called EWISE. Uses Bi-directional LSTM encoders, self-attention, and Wordnet relations	Outperforms Baselines
Ignacio Iacobacci 2016 [40]	Various weighing schemes for embedding techniques have been studied like average, centroid, fractional decay, and exponential decay	The exponential decay strategy outperformed the others
Christian Hadiwinoto et al. 2019	Proposed two new methods for linear projection of hidden layers in Embeddings by layer weighting and gated linear unit	The best results were obtained from linear projection models.

Transformer-based architectures for NLP: The Transformer architecture was introduced in June 2017 [(Vaswani et al., 2017)]. The main aim of the original research was on translation tasks. This was followed by the introduction of several influential models, including:

BERT (Bidirectional Encoded Representation using Transformers): BERT is a pre-trained auto-encoding language model developed by Google in 2018 [(Devlin et al., 2018)]. It is based on transformers that address the drawbacks of LSTM. They are faster, and input text can be processed simultaneously. Since they are bidirectional, the context of words is better learned. In BERT, the meaning is learned from all positions, i.e., from the entire sentence. Multi-head attention and stacked encoders are used in BERT. They work in two phases, namely pre-training and fine-tuning. In the pre-training phase, BERT learns or understands the language features with vast amounts of data, while in the fine-tuning phase, the feed-forward output layer will be replaced with a new set of output layers based on the specific NLP task.

GPT-3 (Generative Pretrained models): It was developed by OpenAI in 2020. The previous versions of GPT-3 are GPT and GPT-2. GPT [(Radford et al., 2018)] was the first transformer-based pre-trained language model. They are auto-regressive transformer models. A bigger and improved version called GPT-2 [(Radford et al., 2019)] was released in 2019. GPT-3, GPT-3, an even bigger version of GPT-2 that is able to perform well on a variety of tasks without the need for fine-tuning. This technique is called zero-shot learning. This is an autoregressive model which is trained with more than 560 GB web corpora which have 175 billion parameters with 96 layers. This has 4 versions namely

Davinci, Curie, Babbage, and Ada, which vary in trainable parameters 175, 13, 6.7, and 2.7 billion.

Some of the general NLP tasks performed by them are given below:

A quantitative and qualitative analysis of language models for word senses with lexical ambiguity was performed by [(Loureiro et al., 2021)]. A dataset named CoarseWSD-20 with 20 ambiguous words was created and the following models were tested for their performance: the models chosen were fasttext base and crawl, fine-tuned BERT base, and large, and Albert XXL. According to their findings, more training instances do not necessarily lead to better performance, and higher polysemy is not an indicator of lower performance. They achieved state-of-the-art performance for the standard datasets in English.

BERT is good at preserving the syntactic and semantic information which are necessary for the process of NLP. [(Tenney et al., 2019)].

A study from Microsoft Cognitive Services Research Group by [(Wang et al., 2021)] et al. describes GPT-3 as a data labeler that combines pseudo labels generated from the GPT-3 framework with human labels. This can decrease the labeling cost to a large extent. Even though the pseudo labels are a bit noisy, the process is cheaper compared to human labeling. In order to do detection, the publicly available language generation and human-generated languages were used. Most of the data contain GPT-2 generated text. To test the heterogeneity of the model, text generated from GPT-3 and GROVER was also used. The performance of this model was measured in terms of accuracy and area under the curve.

Even though these language models have drawn much attention to the use and misuse of automatically generated text. So now there are efforts put in detecting deep fake scenarios also. An attempt to differentiate between human and machine-generated text was proposed by Fröhling et al. [(Fröhling & Zubiaga, 2021)]. A simpler feature-based classifier was used, instead of using expensive large language models.

This work [(Kumar et al., 2019.)] proposes a supervised classification for WSD named EWISE on a continuous sense embedding rather than a discrete labeled space. This model also achieves zero-shot learning by generating seen as well as unseen labels. The working is as follows: The input sequence is converted to contextualized embeddings using BiLSTM and a self-attention layer. They will undergo a dot product operation with sense embeddings. These sense embeddings from the inventory are generated through a BiLSTM definition encoder. The WordNet relations are also used for learning definition encoders. In comparison with F1 scores of infrequent senses, EWISE outperforms the baselines.

This study [(Iacobacci et al., 2016)] was conducted to analyze how word embeddings can be used in WSD, and they performed a deep analysis of how a WSD system considers different strategies for generating pre-trained embeddings. The first strategy studied was concatenation, in which the vectors of the words near to the ambiguous word are concatenated. In the second strategy, called average, the centroid of all the word vectors surrounding the ambiguous word was computed. In the fractional decay strategy, the importance of a word is assumed to be inversely proportional to its distance from the target word. In the exponential decay method, more preference is given to the closer context, and the weighting is performed exponentially. The three main findings drawn from this work are, word embeddings can be used as new features to improve WSD performance, the exponential decay strategy proves to be consistent in high performance and retrofitted features can outperform the SOTA supervised models.

This work leverages various methods for combining embeddings obtained from pre-trained models to improve the performance of word sense disambiguation [(Hadiwinoto et al., 2019)]. The authors utilized the pre-trained embeddings from BERT using the nearest neighbour matching and linear projection of hidden layers for the output sense vector. They have also proposed two additions for the linear projection model, namely layer weighting and gated linear unit, to combine BERT's hidden representation vectors. These WSD techniques are applied to English all-words task, English lexical sample task, and Chinese Ontonotes. The best results were obtained from linear projection models.
BART [(Lewis & et al., 2019)] and T5 (Text-To-Text Transfer Transformer) [(Raffel & et al., 2019)], are two large sequence-to-sequence pretrained models using the same architecture as the original transformer model. Like the previous transformer models, they are self-supervised models. BART is a denoising autoencoder for pretraining sequence-to-sequence model and is particularly useful when fine-tuned for text generation but also works well for other tasks too. T5 is a model that converts all text-based language problems into a text-to-text format.
Instance-Based Learning/ Exemplar-Based or Memory-based Learning: was built by Ng in 1996 [(Ng & Lee, 1996)], utilizes examples to construct the classification model from multiple knowledge sources. The knowledge sources were part of speech tags of neighboring words, morphological form, surrounding words, local collocations, and verb-object syntactic relation. The most popular technique among these is known as the k-Nearest Neighbor algorithm(kNN). In the kNN model, the example sentences are stored with their features, and as new examples are added, they are simply added to the existing model.
In 1997, Hwee Tou Ng [(Ng, 1997)] studied an exemplar-based learning algorithm called PEBLS, in which the number of nearest neighbors, feature weights, exemplar weights, etc., should be set before running the algorithm. The algorithm was evaluated using 10-fold cross-validation to predict the value of k correctly. The algorithm was tested on two test sets named BC50, of the Brown corpus, and WSJ6, of the wall street journal corpus.
Naive Bayes and exemplar-based algorithms are compared by Escudero et al. and found that exemplar-based algorithms perform better [(Escudero et al., 2000)]. It is also reported that exemplar-based models could represent word meaning which can help handle polysemy effectively [(Erk & Padó, 2010)].
Ensemble methods allow combinations of different supervised machine learning techniques. Thus, it will help to overcome the weakness of one approach using the other. Ted Pederson [(Pedersen, 2000)] reported the performance of Naive Bayesian ensembles in 2000. In 2006, Brody et al. [(Brody et al., 2000)] developed an ensemble for unsupervised WSD. They used extended gloss overlap, wordnet similarity, lexical chains, and structural-semantic interconnections for constructing the ensemble. Fig.4. shows major works on ensemble-based WSD.

Different approaches in exemplar techniques are:

Boosting: This method incrementally builds a robust classifier by combining several weak classifiers. Boosting emphasis on the correct classification of labels that are misclassified by the previous classifiers. A standard method used among them is Adaboost. AdaBoost.MH algorithm was used to implement WSD by Gerard Escudero et al. in 2000 [(Escudero et al., 2000.)]. In this work, classical algorithms like Naive Bayes and Exemplar-based learning were compared with Adaboost.MH and found that Adaboost.MH performs better than them. Some techniques to reduce the computational

cost of weak learners were also discussed. They are Frequency filtering, Local frequency filtering, RLM ranking, Lazy boosting, etc. Among them, lazy learning, which preserves all the features, performs better than others.

Ranking: In this method, each classifier provides ranking for the senses of a given word. The sense which maximizes the sum of its ranks from C1, C2, ..., Cn will be chosen as the correct one. C1, C2..., Cn are the first-order classifiers.

Voting: In majority voting, the sense which got maximum votes among the ensembles is taken as the correct sense. Carpuat et al. [(Carpuat et al., 2004)] developed a majority voting-based WSD system using a non-linear kernel PCA. This new ensemble model outperformed the Naive Bayesian and other boosting ensemble models.

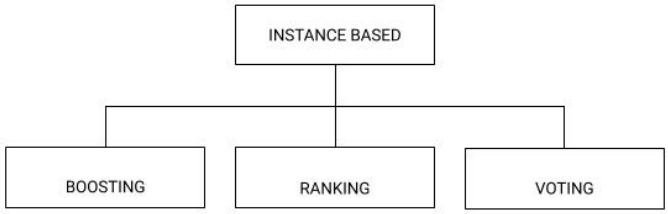


Fig.5. Various instance based WSD techniques

3.1.3 Unsupervised approaches

They do not make use of any type of knowledge sources, and also, they do not assign any labels to data rather group similar words into clusters. They help decrease the knowledge acquisition bottleneck and use "sense discrimination" to separate words into different classes based on their senses.

Typeset sub-subheadings in medium face italic and capitalize the first letter of the first word only. Section numbers to be in roman.

Type-based approaches: Type-based approaches mainly account for context similarity. They are based on the counts of word co-occurrences or associations between words. This is on the assumption that words with similar meanings have related contexts, and they co-occur frequently.

Some of the type-based approaches are given below:

Latent Semantic Analysis (LSA): In 1990, Deerwester et al. [(Deerwester et al., 1990)] proposed a new method for automatic indexing and retrieval of documents known as Latent Semantic Analysis (LSA). LSA aims to improve document retrieval by reducing the term-by-document frequency matrix. This is done using Singular Value Decomposition (SVD). In LSA, the corpus is represented as M X N matrix, where M is the number of word types, and N is the unit of context. Each entry in the cell represents the count of a word that occurs in the context provided by the column.

Hyperspace Analogue to Language (HAL): HAL is a numerical technique that works on the intuition that similar words frequently co-occur [(Burgess et al., 1998)] [(Azzopardi et al., 2005.)]. This method creates a word-by-word co-occurrence matrix by moving a fixed-length window across the vocabulary. For a vocabulary of N words, an N X N matrix is created. HAL is direction sensitive**.**

Clustering By Committee (CBC): This is a descendant of UNICON, which assumes similar words occur in the same contexts [(Pantel, 2003)]. This algorithm takes a word as input and produces clusters that belong to its different senses. The algorithm proceeds in three

steps. In the first phase, each element's similar elements are computed. In the second phase, a collection of clusters called committees are created such that the newly formed committee is not similar to an existing committee. In the last step, each new element is assigned to its most similar clusters.

Determining Predominant Senses: This type-based WSD technique works on the vital idea of inferring a sense to the target word after analyzing the entire text containing the target word. An unsupervised method for WSD by finding word sense predominance was proposed by McCarthy et al. [(McCarthy et al., 2007)]. In their work, an automatically created thesaurus based on Lin's method is used. This gives the k-nearest neighbors of each target word. This score and the Wordnet Semantic Similarity Score (WNSS) are used to find the predominant sense of an ambiguous word.

Rob Koeling and Dian McCarthy [(Koeling & McCarthy, 2007.)] extended this work by considering the coarse-grained sense inventory and the topic-domain dominance. The mapping between coarse sense inventory and the Wordnet is taken into account for generating the thesaurus. Information about the domain-specific word predominance also contributes to some domains.

Extended work on [(Boyd-Graber et al., 2007.)] was done by Jordan Boyd-Graber and David Blei in 2007. They used automatically derived senses by applying Latent Dirichlet Allocation (LDA) with WordNet, where we consider the sense of a random variable is a latent data that can be inferred from the data.

According to Alagic et.al. [(Alagić et al., 2018)], the replacement of ambiguous words with their lexical substitute can improve word sense induction, an unsupervised technique. They used affinity propagation as the unsupervised clustering algorithm that does not know the number of clusters apriori and used measures like CTX, AUTOLS, and a combination of these two to find the linguistic substitute for a given word.

Type-based approaches: Token-based approaches work by clustering context words assuming that the same or related words share the same context. Context group discrimination [(Schütze, 1998)] clusters the contexts in which a word occurs. These methods use second-order co-occurrences that work with Singular value decomposition (SVD) to reduce the dimensionality of the co-occurrence matrix. Pederson and Bruce [(Pedersen & Bruce, 1998)] developed a method in which contexts of target words are grouped based on local features such as POS tags with direction and location, morphology, content collocation, unrestricted collocations, co-occurrences, etc

Co-occurrence graphs: In graph-based algorithms, words are represented as nodes, and the syntactic connections like co-occurrences, co-relations between words as edges. Widdows and Dorow [(Widdows & Dorow, 2002)] proposed a graph-based method for WSD in 2002. In this paper, a graph local to the ambiguous word is built, and an incremental clustering algorithm is applied.

In 2004, Veronis proposed a new algorithm known as HyperLex [(Véronis, 2004.)], which utilizes co-occurrence graphs to find infrequent isolated words using their “small-world” property. Co-occurrence graphs can identify the scarce sense of words by considering the hubs in the graph and can identify high-frequent senses. It has been tested with 10 polysemous words and has shown outstanding performance. Navigli and Lapata [(Navigli & Lapata, 2007)] studied graph connectivity measures for unsupervised WSD in 2007 and came across a conclusion that choosing the sense with the largest degree (Key Player Problem), In degree, and Page rank are the best local measures for WSD. They also found that local measures perform better than global measures.

3.1.4 Structural approaches

Graph-based approach: These methods depend on the structural pattern recognition framework. It would be helpful if we transform the knowledge sources into an organized identifiable fashion. Many graph-based algorithms using Lexical Knowledge Bases (LKB) have evolved in recent years.

Roberto Navigli and Paola Velardi developed a Structural, semantic interconnection method for WSD in 2005 [(Navigli & Velardi., 2005)]. A grammar G is defined that best describes semantic relations like synonymy, hypernymy, meronymy, etc. A semantic graph is created based on the word senses and semantic relations. When a text sequence is given, it first finds the monosemous words and assigns synsets to them. Then the WSD algorithm is applied to all the associations as well as their associated words. Then all ambiguous words in the sequence are identified.

In 2004, Mihalcea et al. [(Mihalcea et al., 2004)] showed that a graph-based page rank algorithm could be used in the context of disambiguation of words. A graph is constructed based on the synsets of WordNet. Page rank algorithm, which was used for web link analysis, was successfully used in WordNet graph with the intuition that when vertex i is linked to vertex j, it is casting a vote for j. The strength of vote from i to j also depends on the rank of i. The more the rank of i, the more the strength of the connection between i and j. Thus, it helps to find the most critical vertex for an ambiguous word from the graph. The Lesk algorithm and a combination of the Lesk+Pagerank algorithm has been experimented for comparison. In this part, the weights of the links joining two synsets are obtained using the Lesk algorithm. Later the work was extended by Rada Mihalcea [(Mihalcea, 2004)] in 2005, Tarau et al. [(Tarau et al., 2005)], Ravi Sinha and Rada Mihalcea [(Sinha & Mihalcea, 2007)] in 2007.

In their study for learning semantic models for WSD, Navigli et al. categorized Wikipedia pages and performed domain-specific WSD. Initially, weighted lists of synsets are created for each domain, which are called Semantic Model Vector (SMV). The authors used a Personalized Pagerank algorithm for finding the most appropriate sense of a word in a particular domain.

In Eneko Agirre et al. [(Agirre & Soroa, 2009)] the authors perform a personalized PageRank algorithm in disambiguation. In this paper, a connected graph is constructed from the WordNet and a personalized PageRank algorithm was used with the words in the context of the input query. Thus, the random walk algorithm gives the context-dependent PageRank.

Complex network approach: Recently, it was reported that when combined with statistical techniques, the complex network approach may be fruitful to enhance the discrimination of senses in large texts. In [(Amancio et al., 2012)], a complex network model of texts found that local structures of these networks could contribute to identifying the disambiguation task. Their results outperformed some of the state-of-the-art systems. The authors used a set of 10 polysemous words from 18 books. They have taken distinct meanings for each word and created a text network, such that a link from i -> j means that the word i appears immediately before j. A count wij is given, which is the number of times the word i appears before the word j.

Repeated nodes in the text are considered the same in the network model. But all the instances of the ambiguous word are taken as a distinct node. Sixteen measurements like degree, strength, length of the shortest path, betweenness centrality, clustering coefficient, etc., were used to analyze the local structure of ambiguous words. They have found that the CN approach is reliable if computed for large networks, and they work better with the k-NN approach.

In 2018, Edilson A. Correa Jr. et al. modeled WSD as a complex network problem [(Correa Jr et al., 2018)]. Their proposed work has considered WSD a classification problem and modeled it using a bipartite heterogeneous network graph. The two layers of nodes contain the feature words (which represent context) and target words (ambiguous words), respectively, and their dependence is used to estimate word senses. The IMBHN algorithm, which could be used for text classification, is used for WSD with gradient descent for error correction.

A new technique that leads to word sense induction was proposed by Correa Jr et.al. [(Corrêa Jr & Amancio, 2019)] very recently. This system makes use of word embeddings using word2vec and applies compositionality rules in it. Google news corpus was used for training, addition, and averaging of word embeddings were performed. Later these vectors will be linked if the context embeddings are similar. They have used both kNN and d-proximity methods for finding the similarity between nodes. Among them, the kNN approach is better in optimizing the modularity of the generated text network. After constructing the context embedding network, the Louvain community detection method is used to identify communities. The communities produced will be treated as different word senses. The authors chose the Louvain Community method because it is known to maintain low computational costs. Some notable works are provided in table IV.

TABLE IV. STRUCTURAL APPROACH TOWARDS WSD

Authors and year	Techniques and features used	Performance or results
Hwee Tou Ng et al. 1996 [67]	They used ensemble methods with knowledge sources like part of speech tags of neighboring words, morphological form, surrounding words, local collocations, and verb-object syntactic relation.	Their work is called LEXAS, tested on the "interest" dataset and corpus acquired from WORDNET, achieving better performance than previous works.
Hwee Tou Ng 1997	Studied an exemplar-based learning algorithm called PEBLS. The features used are the number of nearest neighbors, feature weights, exemplar weights, etc. The algorithm was evaluated using	The algorithm was tested on two test sets named BC50, of the Brown corpus and WSJ6, of the wall street journal corpus. Obtained the highest reported accuracy as equivalent to NB algorithm.

	10-fold cross-validation to predict the value of k correctly.	
Katrin Erk et al. 2010	exemplar-based models could represent word meaning which can help handle polysemy effectively	The experiment was tested on the EP08, EP09, and TDP09 datasets and obtained state-of-the-art accuracy.
Marine CARPUAT et al. 2004	Analyzed that the new Kernel PCA-based model led to accuracy gains in the voting ensemble.	Tested on the Senseval-3 English, Chinese, and Multilingual Lexical Sample tasks. The performance of the voting ensemble increases by adding the KPCA model.
Pederson et.al. 1998	An unsupervised WSD algorithm that uses features such as POStags, morphology, content collocation, unrestricted collocations, co-occurrences etc., which can be extracted from an untagged text dataset. The essential features for the dataset are selected via EM algorithm and Gibbs Sampling method.	Tested on thirteen different words and found that Gibbs sampling produces improved results than the EM algorithm.
Veronis, 2004 [93]	Proposed a new graph-based algorithm known as HyperLex, which utilizes co-occurrence graphs to find infrequent isolated words using their "small-world" property.	It has been tested with 10 polysemous words and shown outstanding performance with a precision of 97%, compared to 73% for baseline tagging and an 82% recall rate
Mihalcea, [59]	Proposed a Knowledge-Based graph page rank algorithm and applied it to WSD. Other approaches like Lesk, most frequent sense, etc. have been combined with page rank algorithms.	Pagerank algorithm alone reduces 21.3% errors with respect to the most frequent sense baseline and a 7.2% error reduction over the Lesk algorithm. Interestingly, combining PageRank and Lesk does not bring any significant improvements over the individual algorithms.
Diego R. Amancio 2012	Studied the relationship between complex network metrics and WSD and found that local structures of these networks could contribute to identifying the disambiguation task. Sixteen measurements like degree, strength, length of the shortest path, betweenness centrality, clustering coefficient, etc., were used to analyze the local structure of ambiguous words.	Their results outperformed some of the state-of-the-art systems. The authors used a set of 10 polysemous words from 18 books. They have found that the CN approach is reliable if computed for large networks, and they work better with the k-NN approach.
Edilson A. Correa Jr. et al. 2018	The authors modeled the data as a bipartite heterogeneous network graph. The two layers of nodes contain the feature words (which represent context) and target words (ambiguous words), respectively, and their dependence is used to estimate word senses. The IMBHN algorithm along with gradient descent is used.	F-score obtained by the the IMBHN in the Senseval-3 exceeded the baseline (MFS) by 8.4% (fine) and 4.4% (coarse) F-score obtained by theIMBHNin the SemEval-2007 along with the baseline

		(More Frequent Sense). The algorithm exceeded the baseline by 5.2%.

3.3. *Other approaches*

3.1.3 Domain-specific WSD:

They disambiguate words with the help of domain knowledge. This is based on the heuristics that the target word will be having a meaning related to its domain. This can be applied in domain-specific applications like tourism, health, sports, architecture, etc. Buitelaar et al [(Buitelaar et al., 2007)] worked on these heuristics and found that they perform with high precision. Domain vectors are created with the help of Wordnet annotations which contain text and sense vectors. Text vector contains the text related to the domain, and sense vector includes the senses of the target word related to the domain. These vectors are used to find the similarity between the senses and the one which scores maximum is chosen. In [(Gliozzo et al., 2004)], the authors proposed a Wordnet, which will be very useful in domain-specific WSD, in which each synset is labeled with domain information.

Cross-lingual WSD: This makes use of information using parallel corpora. This could be helpful to generate sense inventories, and sense tagged corpora. It also focuses on building sense inventories of resource-poor languages with the help of other languages and reducing the complexities of machine translation. SemEval-2010 Task 3 [(Lefever & Hoste., 2013.)] aimed to conduct Cross-Lingual Word Sense Disambiguation of one English noun across five different languages. In this task, five teams participated, and 16 proposals were submitted.

3.1.3 Cross-lingual WSD

This makes use of information using parallel corpora. This could be helpful to generate sense inventories, and sense tagged corpora. It also focuses on building sense inventories of resource-poor languages with the help of other languages and reducing the complexities of machine translation. SemEval-2010 Task 3 [(Lefever & Hoste., 2013.)] aimed to conduct Cross-Lingual Word Sense Disambiguation of one English noun across five different languages. In this task, five teams participated, and 16 proposals were submitted**.**

4. Open Problems in Computational WSD

WSD is considered an AI-complete problem because 100% accuracy is achieved only when computers can think like humans. Here, we list some of the significant problems that WSD approaches usually face.

4.1. *Datasets*

Knowledge-based methods use different knowledge sources such as machine-readable dictionaries or sense inventories, Wikipedia, thesauri, WordNet, ontologies, etc. Various models on knowledge-based WSD are shown in Fig.

4.2. *Context*

Contextual information is vital in WSD; as we change the size of the context window, the performance will get affected. A study based on the size of the micro/ local context window was done by Yarowsky and reported that local ambiguity needs a window of 3-4. In contrast, domain-specific ambiguity requires a larger window. According to Yarowsky, the topical context window also contributes to WSD by exploiting the redundancy in text.

4.3. *Domain*

This approach is based on the underlying concept that the sense of the word is relevant to the current domain. The performance of WSD decreases when this is taken in isolation.

4.4. *Knowledge sources*

All WSD systems rely on knowledge bases, corpora, or machine-readable dictionaries. They work with a considerable amount of data. The percentage of accuracy of WSD increases as the size of the data increases. Lack of extensively annotated corpora is one of the major problems faced by researchers.

5. Evaluation Measures

WSD largely depends on Language since each language is having different usage and sense identification. The evaluation is challenging because of different test sizes, resources used, the set of target words to disambiguate (all words or one-word per sentence), the domain-oriented nature of the text, etc. The evaluation can be done in two ways. WSD is considered a stand-alone, independent application in In-Vitro evaluation. The WSD module is embedded in applications and evaluated in In-Vivo evaluation. Semeval [(Agirre & Soroa, semeval-2007)], [(Moro & Navigli, 2015)], [(Martelli et al., 2021)] and Senseval [(Kilgarri, 1998.)] were some of the benchmarking strategies with evaluation.

Some of the evaluating measures used widely for accessing WSD systems are precision, recall, and F-measure. Precision is also known as a positive predictive value which is defined as the ratio of correct answers provided to the positive answers provided, shown in equation (5). Equation (6) shows recall, also known as sensitivity, the ratio of correct answers provided to the answers provided.

$$Precision = \frac{t_p}{t_p + f_p} \qquad (5)$$

$$Recall = \frac{t_p}{t_p + f_n} \qquad (6)$$

F-measure is the harmonic mean of precision and recall, as shown in equation (7).

$$F - measure = \frac{2(Precision * Recall)}{Precision + Recall} \qquad (7)$$

6. Findings

From this survey, we draw the following findings, which will be helpful for further research.

- Recently a dramatic growth in Word Sense Disambiguation using transformer-based pre-trained language models has been seen. The most popular among them is BERT.
- In recent years complex network models, in which text is represented as a complex network with words being taken as nodes and edges being taken based on adjacency of words, have been used for WSD. The properties of these networks, like degree, strength, shortest paths, betweenness centrality, etc., have been studied for finding the syntactical features of texts.
- There is no gold standard for benchmarking WSD algorithms. The performance of the algorithm depends heavily on the language and availability of resources.
- Most of the reported WSD approaches which had given high accuracy were often used with restricted datasets. Most of the Indian WSD techniques are developed for target word disambiguation.
- Most supervised machine learning algorithms perform better than unsupervised algorithms. But the difficulty in constructing a broad coverage sense tagged training corpora limits the performance of supervised algorithms for Indian Languages.

7. Conclusion and Future Scope

This paper gives a review of various Word Sense Disambiguation techniques developed over time. Even though research in WSD has grown more than 50 years, this field remains dynamic with a great interest for the natural language research community.

The use of WSD as an integrated application is evident, but stand-alone evaluations are still required because of many unresolved problems in WSD. Many unexplored research gaps in the areas of acquisition of linguistic information, combinations of various knowledge sources, domain knowledge, etc., for WSD, in Indian languages could be further investigated in the future.

The analysis of syntactic dependencies at the discourse level, rather than sentence-level, is untouched for our regional languages. Deep learning models like CNN, RNN, and complex networks are not explored much for Indian languages. Research in these directions may bring in better results for WSD

Acknowledgments

The authors would like to thank the management and staff of the Department of Computer Applications, Cochin University of Science and Technology (CUSAT), India for providing infrastructure and support for the conduct of this study.

References

[1] Agirre, E., & Martinez, D. (2000). Exploring automatic word sense disambiguation with decision lists and the Web. arXiv preprint cs/0010024 (2000).

[2] Agirre, E., & Soroa, A. (semeval-2007). Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In Proceedings of the fourth international workshop on semantic evaluations, pp. 7-12. 2007.

[3] Agirre, E., & Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pp. 33-41.

[4] Agrawal, R., & Srikant, R. (1994.). "Fast algorithms for mining association rules." Proc. 20th int. conf. very large data bases, VLDB., Vol. 1215.

[5] Alagić, D., Šnajder, J., & Padó, a. S. (2018.). Leveraging lexical substitutes for unsupervised word sense induction. In the Thirty-Second AAAI Conference on Artificial Intelligence.

[6] Amancio, D. R., Oliveira Jr, O. N., & da F. Costa, L. (2012). Unveiling the relationship between complex networks metrics and word senses. EPL (Europhysics Letters) 98, no. 1, 18002.

[7] Aung, N. T. T., Soe, K. M., & Thein, N. L. (2011). A word sense disambiguation system using Naïve Bayesian algorithm for Myanmar language. International Journal of Scientific & Engineering Research, 2.9, 1-7.

[8] Azzopardi, L., Girolami, M., & Crowe., M. (2005.). Probabilistic hyperspace analogue to language. In Proceedings of the 28th Annual International ACM SIGIR conference on Research and development in information retrieval, pp. 575-576.

[9] Banerjee, S., & Pedersen, T. (2002.). An adapted Lesk algorithm for word sense disambiguation using WordNet. International conference on intelligent text processing and computational linguistics. Springer, Berlin, Heidelberg.

[10] Basile, P., Annalina, C., & Semeraro, G. (2014). An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers.

[11] Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. The journal of machine learning research 3, 1137-1155.

[12] Bhattacharya, I., Getoor, L., & Bengio, Y. (2004.). Unsupervised sense disambiguation using bilingual probabilistic models. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, ACL-04.

[13] Bloehdorn, S., & Hotho, A. (2004). Boosting for text classification with semantic features. International workshop on knowledge discovery on the web. Springer, Berlin, Heidelberg.

[14] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992.). A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory.

[15] Boyd-Graber, J., Blei, D., & Zhu, X. (. 2007.). A topic model for word sense disambiguation. In Proceeding of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 1024-1033.

[16] Brody, S., Navigli, R., & Lapata, M. (n.d.). Ensemble methods for unsupervised wsd. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 97-1.

[17] Buitelaar, P., Magnini, B., Strapparava, C., & Vossen., P. (2007). Domain-specific WSD. In Word Sense Disambiguation, Springer, Dordrecht.

[18] Burgess, C., Livesay, K., & Lund, K. (1998). Explorations in context space: Words, sentences, discourse. Discourse Processes 25, no. 2-3, 211-257.

[19] Carpuat, M., Su, W., & Wu, D. (2004). Augmenting ensemble classification for word sense disambiguation with a Kernel PCA model. In Proceedings of SENSEVAL-3, Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pp. 88-92.

[20] Carpuat, M., & Wu, D. (2005.). Word sense disambiguation vs. statistical machine translation. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05).

[21] Carroll, J., & McCarthy, D. (2000). Word sense disambiguation using automatically acquired verbal preferences. Computers and the Humanities 34.1, 109-114.

[22] Chan, Y. S., Ng, H. T., & Chiang, D. (2007). Word sense disambiguation improves statistical machine translation. Proceedings of the 45th annual meeting of the association of computational linguistics.

[23] Chodorow, M., & Leacock, C. (2000). An unsupervised method for detecting grammatical errors. 1st Meeting of the North American Chapter of the Association for Computational Linguistics.

[24] Choi, Y., Wiebe, a., & Mihalcea, R. (2017). Coarse-grained/-effect word sense disambiguation for implicit sentiment analysis. IEEE Transactions on Affective Computing 8.4, 471-479.

[25] Correa Jr, E. A., A. Lopes, A., & R. Amancio., D. (2018). Word sense disambiguation: A complex network approach. Information Sciences 442, 103-113.

[26] Corrêa Jr, E. A., & Amancio, D. R. (2019). Word sense induction using word embeddings and community detection in complex networks. Physica A: Statistical Mechanics and its Applications 523, pages 180-190.

[27] Cottrell, G. W. (1985.). A connectionist approach to word sense disambiguation. University of Rochester.

[28] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science 41 no. 6, 391-407.

[29] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[30] Elmougy, S., Taher, H., & Noaman, H. (2008.). Naïve Bayes classifier for Arabic word sense disambiguation. proceeding of the 6th International Conference on Informatics and Systems.

[31] Erk, K., & Padó, S. (201). Exemplar-based models for word meaning in context." In Proceedings of the acl 2010 conference short papers. pp. 92-97.

[32] Escudero, G., Marquez, L., & Rigau, G. (2000.). "Boosting applied to word sense disambiguation. In European Conference on Machine Learning, Springer, Berlin, Heidelberg, pp. 129-141.

[33] Escudero, G., Màrquez, L., & Rigau, G. (2000). Naive Bayes and exemplar-based approaches to word sense disambiguation revisited. arXiv preprint cs/0007011.

[34] Fröhling, L., & Zubiaga, A. (2021). Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. PeerJ Computer Science 7, e443.

[35] Gale, W. A., Church, K., & Yarowsky, D. (1992). One sense per discourse. Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February, 23-26.

[36] Gale, W. A., Church, K. W., & Yarowsky, D. (1992). A method for disambiguating word senses in a large corpus. Computers and the Humanities 26.5, 415-439.

[37] Gelbukh, A. (2014). Word sense disambiguation through associative dictionaries. Diss. INSTITUTO POLITÉCNICO NACIONAL.

[38] Gliozzo, A., Strapparava, C., & Dagan, I. (2004). Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. Computer Speech & Language 18, no. 3, 275-299.

[39] Hadiwinoto, C., Tou Ng, H., & Chung Gan, W. (2019). Improved word sense disambiguation using pre-trained contextualized word representations. arXiv preprint arXiv:1910.00194.

[40] Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2016, August 7-12). Embeddings for word sense disambiguation: An evaluation study." In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, Volume 1: Long Papers, 897–907.

[41] Ide, N. (2000). Cross-lingual sense determination: Can it work? Computers and the Humanities 34.1, 223-234.

[42] Ide, N., Erjavec, T., & Tufis, D. (2002, July). Sense Discrimination with Parallel Corpora" in Proceedings of the SIGLEX Workshop on Word Sense Disambiguation: Recent Successes and Future Directions. ACL2002, Philadelphia, 56-60.

[43] Joshi, M., Pedersen, T., & Maclin, R. (2005). A Comparative Study of Support Vector Machines Applied to the Supervised Word Sense Disambiguation Problem in the Medical Domain. IICAI.

[44] Kågebäck, M., & Salomonsson, H. (2016). Word sense disambiguation using a bidirectional lstm. arXiv preprint arXiv:1606.03568.

[45] Kilgarri, A. (1998.). Senseval: An exercise in evaluating word sense disambiguation programs. In Proc. of the first international conference on language resources and evaluation, pp. 581-588.

[46] Koeling, R., & McCarthy, D. (2007.). Sussx: WSD using automatically acquired predominant senses. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pp. 314-317.

[47] Kumar, S., Jat, S., Saxena, K., & Talukdar, P. (2019.). Zero-shot word sense disambiguation using sense definition embeddings. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5670-5681.

[48] Le, C. A., & Shimazu, A. (2004). High WSD accuracy using Naive Bayesian classifier with rich features. Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation.

[49] Lee, Y. K., & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002).

[50] Lee, Y. K., Ng, H. T., & Chia., T. K. (2004.). Supervised word sense disambiguation with support vector machines and multiple knowledge sources. Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.

[51] Lefever, E., & Hoste., V. (2013.). Semeval-2013 task 10: Cross-lingual word sense disambiguation. " In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pp. 158-166.

[52] Lesk, M. (1986, June). 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 (pp. 24-26).

[53] Lewis, M., & et al. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[54] Loureiro, D., Rezaee, K., Pilehvar, M. T., & Camacho-Collados., J. (2021). Analysis and Evaluation of Language Models for Word Sense Disambiguation. Computational Linguistics: 1-55.

[55] Luo, F., Liu, T., He, Z., Xia, Q., Sui, Z., & Chang, B. (2018.). Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing., pp. 1402-1411.

[56] Martelli, F., Kalach, N., Tola, G., & Navigli, R. (2021). SemEval-2021 Task 2: Multilingual and Cross-lingual Word-in-Context Disambiguation (MCL-WiC)." In Proceedings of the Fifteenth Workshop on Semantic Evaluation (SemEval-2021).

[57] McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2007). Unsupervised acquisition of predominant word senses. Computational Linguistics 33, no. 4, 553-590.

[58] Mihalcea, R. (2004). Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2005, pp. 411–418.

[59] Mihalcea, R., Paul, T., & Figa, E. (2004). PageRank on semantic networks, with application to word sense disambiguation. In COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pp. 1126-1132.

[60] Mikolov, T., Sutskever, I., Chen, K., S. Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pp. 3111-3119.

[61] Moro, A., & Navigli, R. (2015). Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking." In Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp. 288-297.

[62] Navigli, R. (2009). Word sense disambiguation: A survey. ACM computing surveys (CSUR) 41, no. 2, 1-69.

[63] Navigli, R., & Lapata, M. (2007). Graph connectivity measures for unsupervised word sense disambiguation. In IJCAI, vol. 7, pp. 1683-1688.

[64] Navigli, R., & Velardi., P. (2005). Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. IEEE transactions on pattern analysis and machine intelligence 27, no. 7, 1075-1086.

[65] Ng, H. T. (1997). Exemplar-based word sense disambiguation: Some recent improvements. arXiv preprint cmp-lg/9706010.

[66] Ng, H. T. (1997). Getting serious about word sense disambiguation. Tagging Text with Lexical Semantics: Why, What, and How?

[67] Ng, H. T., & Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. arXiv preprint cmp-lg/9606032.

[68] Pantel, P. A. (2003). Clustering by committee. Alberta, Canada: University of Alberta.

[69] Pasca, M. A., & Harabagiu., S. M. (2001). High performance question/answering. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval.

[70] Patwardhan, S., Banerjee, S., & Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. International conference on intelligent text processing and computational linguistics. Springer, Berlin, Heidelberg.

[71] Pedersen, T. (2000). A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. arXiv preprint cs/0005006.

[72] Pedersen, T. (2001). A decision tree of bigrams is an accurate predictor of word sense. arXiv preprint cs/0103026.

[73] Pedersen, T., & Bruce, R. (1998). Knowledge lean word-sense disambiguation. In AAAI/IAAI, pp. 800-805.

[74] Pennington, J., Socher, R., & D. Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543.

[75] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[76] Poluru, S. (2021.). Word Sense Disambiguation for Telugu Using Lesk. Machine Learning Technologies and Applications: Proceedings of ICACECS 2020. Springer Singapore.

[77] Quinlan, R. J. (1986). Induction of decision trees. Machine learning 1.1, 81-106.

[78] Radford, A., Jeffrey, W., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog 1, no. 8 (2019): 9.

[79] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[80] Raffel, C., & et al. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv preprint arXiv:1910.10683.

[81] Raganato, A., Bovi, C. D., & Navigli, R. (2017.). Neural sequence learning models for word sense disambiguation. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.

[82] Resnik, P. (1997). Selectional preference and sense disambiguation. Tagging Text with Lexical Semantics: Why, What, and How?

[83] Rivest, R. L. (1987). Learning decision lists. Machine learning 2.3, 229-246.

[84] Rumelhart, D. E., Hinton, G. E., & J. Williams, R. (1986). Learning representations by back-propagating errors. nature 323 no. 6088, 533-536.

[85] Schütze, H. (1998). Automatic word sense discrimination. Computational linguistics 24, no. 1, 97-123.

[86] Seneff, S. (1992). TINA: A natural language system for spoken language applications. Computational linguistics 18.1, 61-86.

[87] Singh, S., Rauniyar, R., & Manohar., M. (2021.). Nepali Word-Sense Disambiguation Using Variants of Simplified Lesk Measure. Data Science. Springer, Singapore, 41-57.

[88] Sinha, R., & Mihalcea, R. (2007). Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In International conference on semantic computing (ICSC 2007), IEEE, pp. 363-369.

[89] Sun, X.-R., Lv, S.-H., Wang, X.-D., & Wang, D. (n.d.). Chinese word sense disambiguation using a LSTM. In ITM Web of Conferences, EDP Sciences, 2017., vol. 12, p. 01027.

[90] Tarau, P., Mihalcea, R., & Figa, E. (2005). Semantic document engineering with WordNet and PageRank. In Proceedings of the 2005 ACM symposium on Applied computing, pp. 782-786.

[91] Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. arXiv preprint arXiv:1905.05950.

[92] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin., I. (2017). Attention is all you need. Advances in neural information processing systems, pp. 5998-6008.

[93] Véronis, J. (223-252.). Hyperlex: lexical cartography for information retrieval. Computer Speech & Language 18, no. 3, 2004.

[94] Veronis, J., & Ide, N. (1990). Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics.

[95] Wang, S., Liu, Y., Xu, Y., Zhu, C., & Zeng, M. (2021). Want To Reduce Labeling Cost? GPT-3 Can Help. arXiv preprint arXiv:2108.13487.

[96] Widdows, D., & Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In COLING 2002: The 19th International Conference on Computational Linguistics.

[97] Yarowsky, D. (1994). DECISION LISTS FOR LEXICAL AMBIGUITY RESOLUTION: Application to Accent Restoration in Spanish and French. arXiv preprint cmp-lg/9406034.

[98] Yarowsky, D. (2000, Apr). Hierarchical Decision Lists for Word Sense Disambiguation. Computers and the Humanities, Vol. 34, 179-186.

[99] Zhong, L. Y., & Wang., T. H. (2020). Towards word sense disambiguation using multiple kernel support vector machines. International Journal of Innovative Computing, Information and Control, 16(2), 555-570.

[100] Zouaghi, A., Merhbene, L., & Zrigui, M. (2012). Combination of information retrieval methods with LESK algorithm for Arabic word sense disambiguation. Artificial Intelligence Review 38.4 (2012): 257-269., 38.4 (2012): 257-269.

[101] https://towardsdatascience.com/tutorial-on-lstm-a-computational-perspective-f3417442c2cd