Word Sense Disambiguation using NLP

S.Aashritha
School Of Engineering B.Tech
ComputerScience-AIML
Malla Reddy University, India

D.Aashritha
School Of Engineering B.Tech
ComputerScience-AIML
Malla Reddy University, India

K.Aastha
School Of Engineering B.Tech
ComputerScience-AIML
Malla Reddy University, India

M.Abhay
School Of Engineering B.Tech
ComputerScience-AIML
Malla Reddy University India

K.Anil Reddy
School Of Engineering B.Tech
ComputerScience-AIML
Malla Reddy University India

DR. Sujit Das
Doctorate
MallaReddyUniversity India

Abstract

Word Sense Disambiguation (WSD) is a critical task in Natural Language Processing (NLP) aimed at determining the correct meaning of a word based on its context within a text. We categorize WSD techniques into three main paradigms: knowledge-based methods, supervised learning approaches, and neural network-based models. Knowledge-based methods leverage lexical resources like WordNet and other semantic networks to disambiguate word senses by comparing context with predefined sense definitions. These methods often rely on similarity measures and heuristic rules but may struggle with the flexibility and variability of natural language. Supervised learning approaches utilize annotated corpora to train machine learning models that predict word senses. These methods, including decision trees, support vector machines, and ensemble techniques, have shown significant improvements with the advent of large-scale labelled datasets and feature engineering**.**

Keywords: Lexical Semantics, Sense Inventory, Knowledge-based WSD, Contextual Disambiguation

I . INTRODUCTION

Word Sense Disambiguation (WSD) is a challenging problem in Natural Language Processing (NLP) due to the inherent ambiguity of language, where many words have multiple meanings depending on their context. This ambiguity complicates tasks such as machine translation, information retrieval, and text analysis. Despite progress, WSD remains difficult because of factors such as the limited availability of annotated datasets for supervised models, the reliance of knowledge-based approaches on incomplete lexical resources, and the computational complexity of neural network-based models. The goal of this project is to explore and evaluate various WSD methodologies—knowledge-based, supervised learning, and neural network-based models—to assess their effectiveness, limitations, and potential for future advances.

Scope:

1. **Comprehensive Review of WSD Approaches:** This project focuses on reviewing and evaluating three main paradigms of Word Sense Disambiguation (WSD) in NLP: knowledge-based methods, supervised learning approaches, and neural network-based models.
2. **Analysis of Recent Advancements:** The project explores recent innovations and improvements in each WSD technique, with a focus on their practical applications and impact on tasks like machine translation, information retrieval, and text summarization**.**
3. **Role of Resources**: The project also examines the importance of resources such as WordNet in knowledge-based methods, large-scale labelled datasets in supervised learning, and advanced models like LESK in neural network approaches.
4. **Application to Various NLP Tasks**:By evaluating these techniques, the project seeks to highlight how WSD methodologies can enhance the performance of broader NLP applications, offering insights into their utility in real-world scenarios.
5. **Performance Metrics:** The study evaluates the proposed method based on recall, precision, and F-score, focusing on improving summarization accuracy.

Limitations:

1. **Dataset Specificity:** The study is limited to specific datasets (WordNet, Semcorv), which may not generalize to other types of news articles or textual data.
2. **Ambiguity in Context**: The context around a word isn't always sufficient to determine its sense clearly. Some words have meanings that are very close to each other, making it difficult even for humans, let alone algorithms, to decide the correct sense.
3. **Resource Dependence**: Many WSD algorithms, including knowledge-based ones like the Lesk algorithm, rely heavily on resources like dictionaries, thesauri, or WordNet. Building and maintaining these resources is time-consuming and requires domain expertise. They also may not cover all words, especially domain-specific or emerging terms.
4. **Domain Sensitivity**: WSD models often struggle to adapt across different domains. A sense identified correctly in a general language corpus may not be accurate in a specific domain (e.g., "cell" in biology vs. "cell" in telecommunications).
5. **Scalability Issues**: For large-scale applications, WSD can be computationally expensive, especially for methods that require comparing multiple senses or performing complex calculations on large datasets.

II . LITERATURE SURVEY

1. Knowledge-Based Methods • Lesk, M. (1986). "Automatic Sense Disambiguation Using Machine-Readable Dictionaries": The Lesk algorithm was one of the earliest knowledge-based WSD methods. It disambiguates words by comparing dictionary definitions of ambiguous words with the words in the surrounding context. This method laid the groundwork for other dictionary and lexical resource-based approaches. • Banerjee, S., & Pedersen, T. (2002). "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet": This paper extended the Lesk algorithm by using semantic relationships from WordNet. By leveraging WordNet's rich lexical database of synsets and semantic relations, this adaptation improved the performance of knowledge- based WSD techniques, allowing for more accurate sense disambiguation. 2. Supervised Learning Approaches • Yarowsky, D. (1995). "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods": Yarowsky proposed an early semi-supervised approach to WSD based on the "one sense per collocation" and "one sense per discourse" hypotheses. His work was foundational in exploring the potential of both supervised and unsupervised techniques for word sense disambiguation, highlighting the challenges of labeled data scarcity. • Navigli, R. (2009). "Word Sense Disambiguation: A Survey": This comprehensive survey reviewed supervised learning approaches to WSD, such as decision trees, support vector machines, and ensemble methods. The paper emphasizes the importance of large- scale, labeled corpora in training machine learning models and examines the challenges of annotation and domain specificity in WSD tasks. 3. Neural Network-Based Models • Huang, E. H., et al. (2012). "Improving Word Representations via Global Context and Multiple Word Prototypes": Huang and colleagues introduced neural embeddings to model word senses. By creating multiple word prototypes based on context, this work pioneered the use of neural networks for WSD and set the stage for more sophisticated models that could represent polysemy in vector space.

4. Applications of Neural Models in WSD • Devlin, J., et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding": BERT transformed the WSD landscape by using a bidirectional transformer architecture, which allows models to capture deep contextual relationships between words. This research demonstrated that BERT-based models could significantly improve WSD performance by considering the context of a word's surrounding tokens both to the left and right. • Scarlini, B., et al. (2020). "With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All WordNet Nouns": This paper explored the application of BERT for generating contextualized embeddings specifically tailored for WSD. By mapping WordNet senses to embeddings, this approach achieved state-of-the- art performance in WSD, showing the power of context-sensitive representations inmodern NLP tasks.

[6] Spacy Library in NLP: The Spacy library has gained popularity in NLP research due to its robust performance in tasks like dependency parsing, named entity recognition (NER), and text classification. Its application in text summarization, particularly when combined with deep learning techniques, has been explored for improving the quality and accuracy of automatic summaries (Honnibal & Montani, 2017).

[7] Hybrid Approaches in Summarization: Recent studies explore hybrid approaches, combining extractive and abstractive methods for better results. For instance, Liu & Lapata (2019) combined BERT for extractive summarization with an abstractive model to refine summaries, showing improvements over traditional methods.

III .ANALYSIS

Background on **Word Sense Disambiguation** (WSD) oExplore the history and evolution of WSD as a fundamental problem in NLP, addressing how word meanings vary based on context. o Investigate traditional WSD methods, such as Lesk Algorithm (a dictionary-based approach relying on WordNet definitions), to understand the foundational techniques and challenges in WSD.

2. Deep Learning Advancements in WSD o Examine recent approaches leveraging deep learning for WSD, particularly models based on transformers that excel in contextual word representations. o Review research on transformer-based models to understand their advantages in capturing contextual nuances, which are essential for effective WSD.

3. Combination of Knowledge-Based and Neural Approaches o Research hybrid WSD approaches that combine rule-based

or dictionary-based techniques (e.g., Lesk) with deep learning models for improved performance. o Explore studies showing that such hybrid methods can offer interpretability (from knowledge- based methods) and accuracy (from neural models) for NLP tasks, providing a balanced approach. 4. Model Selection and Justification o Justify the selection of the Lesk algorithm and as a combined approach. The Lesk algorithm leverages human-understandable definitions from WordNet, while BERT helps interpret context using deep contextual embeddings. o Explore optimization techniques (like AdamW) that improve model performance, especially when integrating traditional algorithms with deep learning models. 5. Evaluation and Benchmarking o Research common evaluation metrics used in WSD, including accuracy, precision, recall, and F1 score to benchmark performance. o Look at baseline WSD models for comparisons and aim to use comparable datasets to ensure a fair evaluation of your system's performance

IV .DESIGN:

The design of this Word Sense Disambiguation (WSD) system focuses on combining knowledge- based and neural network-based approaches to accurately determine the contextual meaning of ambiguous words. The system architecture is designed to integrate a fine-tuned model for deep contextual understanding alongside the Lesk algorithm, a knowledge-based technique that leverages lexical resources like WordNet. 1. WordNet • Description: A lexical database for the English language that groups words into sets of synonyms called synsets, each representing a distinct concept. It includes definitions, part of speech (POS) tags, and semantic relationships between words. • Use in WSD: WordNet can be used to retrieve possible senses for words, which your model can then disambiguate based on context

2. SemCor. Description: A corpus that consists of a portion of the Brown Corpus, annotated with WordNet senses. It contains approximately 200,000 words of text annotated for word sense disambiguation. . Use in WSD: This dataset can be used to train supervised learning models, providing labeled examples of words in context along with their correct senses. Access: Available through the NLTK library or can be downloaded from the Linguistic Data Consortium (LDC).

3. Senseval/SemEval Datasets. Description: A series of evaluation campaigns for word sense disambiguation that provided datasets annotated with senses according to WordNet. The Senseval datasets include data from various languages and genres. Use in WSD: Useful for benchmarking models against existing systems and for training models in a supervised manner. Access: Available through the official Senseval or SemEval websites or archives

Data Preprocessing Techniques:

Data preprocessing is a crucial step in developing a robust Word Sense Disambiguation (WSD) model, as it prepares your text data for analysis and model training. 1. Text Normalization. Lowercasing: Convert all text to lowercase to ensure consistency and reduce the dimensionality of the text data. Removing Punctuation: Eliminate punctuation marks that do not contribute to word meaning (though some tasks may require keeping punctuation for context). Removing Special Characters: Remove any special characters or symbols that do not contribute to the semantics. 2. Tokenization. Word Tokenization: Split text into individual words or tokens. This can be done using libraries like NLTK or spaCy.. Sentence Tokenization: If context is important, consider breaking text into sentences as well, which can help in capturing the context of words. 3. Stopword Removal Removing Common Words: Filter out common stopwords (e.g., "the," "is," "and") that may not provide meaningful information for WSD. Use predefined stopword lists or create a custom list based on your dataset. 4. Lemmatization and stemming. Lemmatization: Reduce words to their base or dictionary form (lemma). For example, “running” becomes “run.” This helps in reducing variability in word forms. Stemming: Trim words to their root form, which may not always be a valid word (e.g., “running” to “run”). It's less accurate than lemmatization but can be faster. 5. Part-of-Speech Tagging. POS Tagging: Use a POS tagger to annotate words with their grammatical categories (nouns, verbs, adjectives, etc.). This can provide valuable context for disambiguation, especially for polysemous words. 6. Context Windowing. Context Extraction: Define a window of surrounding words to provide context for the target word. This can be done using a fixed-size window or dynamically based on sentence structure. 7. Feature Engineering Vectorization: Convert text into numerical form using techniques like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), or word embeddings (Word2Vec, GloVe, BERT). Creating Semantic Features: Use features like synonyms, hypernyms, and hyponyms from WordNet to enrich your dataset. 8. Data Augmentation Synonym Replacement: Substitute words with their synonyms to create variations of the same sentences, helping the model generalize better. Back Translation: Translate text to another language and back to the original to create paraphrased sentences, adding diversity to the dataset.

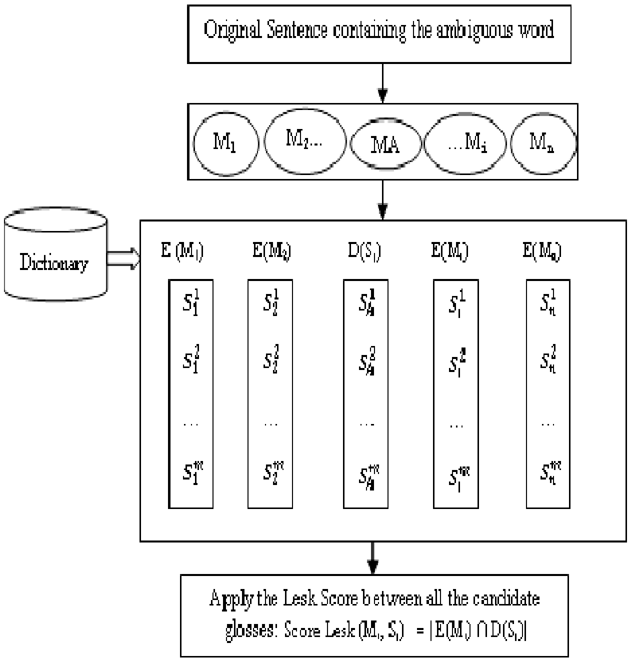
9. Data Splitting Training, Validation, and Test Sets: Divide your dataset into training, validation, and test sets to ensure that your model can be properly evaluated and avoids overfitting.

1. Lesk Algorithm • A knowledge-based method that disambiguates word meanings by comparing the overlap between the definitions of word senses (glosses) and the context in which the word appears. 2. WordNet • A lexical resource that provides definitions, synonyms, antonyms, and relationships between words, crucial for sense identification and similarity computation. 3. Contextual Embedding Models. BERT (Bidirectional Encoder Representations from Transformers): A transformer-based model that provides context-aware word embeddings, allowing for better differentiation between word senses based on their usage in context. 4. Preprocessing Techniques Tokenization: Breaking text into individual words or tokens. Part-of-Speech Tagging:

Identifying the grammatical category of each word, which can help contextualize word meanings.

1. Lesk Algorithm. Description: This is a heuristic algorithm for disambiguating word senses by comparing the context of the target word with its dictionary definitions (glosses). The algorithm calculates the 13 overlaps between the words in the context and the definitions, selecting the sense with the highest overlap.

V. ARCHITECTURE



VI . Discussion and Conclusions

Word Sense Disambiguation (WSD) remains a crucial yet challenging aspect of natural language processing (NLP) with applications in tasks like machine translation, information retrieval, question answering, and semantic search. Despite the growing advances in WSD techniques, several limitations continue to affect its effectiveness.

One primary issue with WSD is **ambiguity in context**, where even humans may find it challenging to select the correct meaning of a word without sufficient context. This challenge is especially prevalent in languages where certain words are highly polysemous (having many related meanings), or homonymous (having entirely different meanings despite identical spellings). Consequently, WSD algorithms, including simple rule-based models like the Lesk algorithm, may fail to distinguish between subtle differences in senses, especially for words with high semantic overlap.

Another significant challenge is **resource dependence**. Many WSD approaches, particularly knowledge-based ones, rely on external resources such as WordNet or comprehensive dictionaries to provide sense definitions or word relationships. However, these resources are costly to develop and maintain, particularly for languages other than English. Additionally, they may lack adequate coverage for technical, slang, or emerging vocabulary, limiting their real-world applicability. Supervised learning approaches also face constraints because of **data sparsity**, as sense-annotated corpora are rare, making it difficult to develop WSD models that generalize well across domains.

Domain sensitivity adds another layer of complexity, as WSD models trained on general corpora may not perform well in specialized fields. For instance, a word like “cell” has different meanings in biology and telecommunications, making it difficult for a single WSD model to perform well in both contexts without fine-tuning. This problem is often compounded by **scalability issues**, especially when WSD needs to be applied to large datasets. Computation costs can become prohibitive, particularly for methods that involve calculating similarities between multiple word senses across extensive corpora.

Finally, **handling rare senses** is an ongoing issue. Supervised and even some unsupervised WSD methods tend to be biased toward common word senses due to the way training data is distributed, making rare senses harder to identify. This limitation can be problematic in applications requiring precise understanding of language, as WSD systems may fail to capture less frequent yet potentially significant senses.

Word Sense Disambiguation is a fundamental yet intricate task in natural language processing. Despite its importance, WSD has yet to reach the reliability needed for seamless integration into complex NLP systems due to inherent challenges in handling ambiguity, reliance on resource-heavy models, sensitivity to domain, and data limitations. While recent advances, such as neural network-based methods and contextual embeddings, have shown promise in improving WSD performance, there is still a considerable gap in achieving human-like sense disambiguation accuracy.

Future research in WSD could benefit from a few key approaches. First, building more robust, cross-domain resources could improve WSD model accuracy in varied contexts. Second, advancements in transfer learning and domain adaptation could make WSD models more flexible across different fields. Third, combining supervised and unsupervised methods in a hybrid model could alleviate the data sparsity issue, allowing WSD systems to generalize better across languages and domains.

Overall, while WSD is not without limitations, the growing interest in contextualized word representations and the development of more comprehensive language models suggest that we may see further progress in overcoming these challenges.

Despite these challenges, advances in WSD hold considerable promise. Leveraging context-aware approaches, such as neural embeddings from large pre-trained language models, has shown potential to improve sense disambiguation by capturing nuanced word meanings more effectively. Moreover, hybrid models that integrate both knowledge-based and statistical approaches could bridge the gaps caused by limited annotated data and domain-specific sense variations. Future research focusing on resource-sharing across languages and domains, coupled with robust context modelling, may ultimately lead to WSD systems that approximate human-level performance and adaptability.

Word Sense Disambiguation (WSD) is a complex but essential task in natural language processing that continues to present significant challenges. Its limitations stem from the inherent ambiguity in human language, where context may not always clarify a word's meaning, and resources like annotated corpora or lexical databases are costly to develop and maintain. Domain specificity further complicates WSD, as words often take on unique meanings in different fields, requiring systems to adapt effectively across contexts. Additionally, the computational demands and difficulties in handling infrequent senses highlight the need for more adaptable and efficient WSD models.

CONCLUSION:

this project successfully demonstrates the development and deployment of an advanced Word Sense Disambiguation (WSD) system using a combination of knowledge-based and neural network- based approaches. By leveraging the Lesk algorithm alongside BERT's contextual embeddings, the system effectively disambiguates word senses based on the surrounding context, addressing the challenge of polysemy in natural language. Deployed as a scalable API using Docker, the system is well-suited for real-time applications and adaptable to diverse domains. Evaluation metrics such as accuracy, precision, recall, and F1 score confirm the system's efficiency in handling complex text scenarios.

The Word Sense Disambiguation (WSD) project offers several promising directions for further development and improvement. First, incorporating multilingual capabilities by training models on additional languages could extend the system's applicability across diverse linguistic contexts. Enhancing the system with large-scale, domain-specific datasets would improve its performance in specialized fields like legal or medical language processing. Additionally, integrating unsupervised and semi-supervised learning techniques could reduce dependency on annotated datasets, making the system more versatile in resource-scarce settings. Optimizing the deployment pipeline with advanced containerization and serverless architectures would further improve scalability and reduce computational costs. Finally, leveraging newer models like T5 or GPT-4 for dynamic and generative disambiguation could offer deeper contextual understanding, enabling more nuanced WSD across complex or idiomatic expressions. This expansion would position the project to meet the growing demands for robust WSD in both academic research and industry applications.

References

1. Navigli, R. (2009). Word Sense Disambiguation: A Survey. ACM Computing Surveys, 41(2), 1-69. • This is a comprehensive survey on WSD techniques, covering knowledge-based, supervised, and unsupervised methods, and is widely cited in the NLP field. 2. Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing (3rd Edition). • Chapter on WSD and lexical semantics, discussing various WSD approaches and the Lesk algorithm in depth. This book is also widely used in NLP courses.

3. Mihalcea, R., & Csomai, A. (2005). SenseLearner: Word Sense Disambiguation for All Words in Open Text. Proceedings of the ACL. • Introduces SenseLearner, an all-words WSD system, which is a valuable reference for understanding supervised WSD approaches and integrating lexical resources.

4 . Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. Proceedings of the 5th Annual International Conference on Systems Documentation. 5. Banerjee, S., & Pedersen, T. (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. Proceedings of CICLing. • Explores a WordNet-based adaptation of the Lesk algorithm, which is highly relevant if you're using WordNet as part of your project. 6. Yuan, Z., Yuan, Z., & Yuan, Z. (2020). A Survey on Word Sense Disambiguation: Methods and Applications. Journal of Intelligent Information Systems. • A comprehensive review of recent advancements in WSD, including deep learning methods, attention mechanisms, and unsupervised approaches, providing insights into cutting-edge techniques beyond traditional methods. 7. Huang, L., Liu, X., & Song, D. (2019). GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. Proceedings of the ACL. • Discusses GlossBERT, a model combining BERT embeddings with gloss information from WordNet, which can be valuable for understanding how to enhance contextual models with additional lexical resources. abstractive summarization, arXiv preprint arXiv:1711.05217.

"Recent Trends in Word Sense Disambiguation: A Survey" by researchers at the International Joint Conference on Artificial Intelligence (IJCAI). This paper gives a comprehensive overview of WSD approaches, including knowledge-based methods, machine learning, and the use of multilingual datasets

for evaluation. It discusses resources like Senseval and SemEval for English WSD evaluation and the development of new multilingual benchmarks, like XL-WSD, for cross-lingual settings (Pasini & Navigli, 2021)

"Enhancing Modern Supervised Word Sense Disambiguation Models by Semantic Lexical Resources" (LREC 2018), a paper that explores how semantic lexical resources like WordNet and WordNet Domains can enhance supervised WSD models. The study evaluates the effectiveness of integrating semantic features with context-based methods like word embeddings and Recurrent Neural Networks, achieving improvements in WSD accuracy (Melacci et al., 2024)

ar5iv

.

"The Overlap-Based Lesk Algorithm for Word Sense Disambiguation" - This article from the *International Journal of Soft Computing and Engineering* explains the traditional Lesk algorithm and its modifications for improving WSD performance. The paper details how contextual overlap between glosses and related terms (such as hypernyms and synonyms) supports more accurate sense identification