Integrating Hierarchical attention and Context-aware embedding for improved Word sense disambiguation performance using BiLSTM model.

By

Robbel Habtamu Yigzaw

A Thesis Submitted to the Graduate School of Information Technology & Engineering, Addis Ababa Institute of Technology, Addis Ababa University In Partial Fulfillment of the Requirements for the Degree of Master of Science in Artificial Intelligence

Supervised by: Dr. Beakal Gizachew Assefa

Addis Ababa, Ethiopia

June, 2024

APPROVAL

This is to certify that the thesis entitled **"Integrating Hierarchical attention and Context-aware embedding for improved Word sense disambiguation performance using BiLSTM model."** submitted by **Robbel Habtamu Yigzaw** in partial fulfillment of the thesis-option requirements for the Degree of Master of Science in Artificial Intelligence at the School of Information Technology & Engineering, Addis Ababa Institute of Technology, has been examined and is recommended for acceptance and approval.

Advisor:

Name: Beakal Gizachew (Ph.D) Signature: Date: 11-June-2024

External Examiner:

Name: Solomon T. (PhD) Signature: Date: 11/06/2024

Internal Examiner:

Name: Fantahun B. (PhD) Signature: Date: 11-JUN-2024

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor, **Dr. Beakal Gizachew**, for his invaluable guidance, support, and encouragement throughout this thesis research. His expertise and insights have been instrumental in shaping this work and my academic growth. This thesis would not have been achievable without his continuous support, supervision, feedback, and the freedom to explore various approaches.

I am also deeply grateful to **my family** for their unwavering love, understanding, and support throughout my academic journey. Their encouragement and belief in me have been a constant source of motivation. I would also like to extend my heartfelt thanks to my friends **Chala, Azmeraw, Yosef, and Lakew** for their encouragement, understanding, and for being a source of joy and laughter during challenging times. Their friendship has made this journey memorable and meaningful.

DEDICATION

...To My Self...

ABSTRACT

Word Sense Disambiguation is a fundamental task in natural language processing, aiming to determine the correct sense of a word based on its context. Word sense ambiguity, such as polysomy, and semantic ambiguity poses significant challenges in the task of WSD. Recent advancements in research have focused on utilizing deep contextual models to address these challenges. However, despite this positive progress, semantical ambiguity remains a challenge, especially when dealing with polysomy words. This research introduces a new approach that integrates hierarchical attention mechanisms and BERT embeddings to enhance WSD accuracy. Our model, incorporating both local and global attention, demonstrates significant improvements in accuracy, particularly in complex sentence structures. To the best of our knowledge, our model is the first to incorporate hierarchical attention mechanisms integrated with contextual embedding. This integration enhances the model's performance, especially when combined with the contextual model BERT as word embeddings. Through extensive experimentation, we demonstrate the effectiveness of our proposed model. Our research highlights several key points. First, we showcase the effectiveness of hierarchical attention and contextual embeddings for WSD. Second, we adapted the model to Amharic word sense disambiguation, demonstrating strong performance. Despite the lack of a standard benchmark dataset for Amharic WSD, our model performs 92.4% Accuracy on a self-prepared dataset. Third, our findings emphasize the importance of linguistic features in capturing relevant contextual information for WSD. We also note that Part-of-Speech (POS) tagging has a less significant impact on our English data, while word embeddings significantly impact model performance. Furthermore, applying local and global attention leads to better results, with local attention at the word level showing promising results. Overall, our model achieves state-of-the-art results in WSD within the same framework. Our results demonstrate a significant improvement of 1.8% to 2.9% F1 score over baseline models. We also achieve state-of-the-art performance on the Italian language by achieving 0.5% to 0.7% F1 score over baseline papers. These findings underscore the importance of considering contextual information in WSD, paving the way for more sophisticated and context-aware natural language processing systems.

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AI Artificial Intelligence

AmRoBERTa Amharic RoBERTa

BERT Bidirectional Encoder Representations from Transformers

BiLSTM Bidirectional Long Short-Term Memory

CNNs Convolutional Neural Networks

CPU Central Processing Unit

CRF Conditional Random Field

CSV comma-separated values

DSR Design Science Research

ELMo Embeddings from Language Models

GloVe Global Vectors for Word Representation

GRU Gated Recurrent Unit

GPU Graphics Processing Unit

HMM Hidden Markov Model

LSTM Long Short-Term Memory

NLTK Natural Language Toolkit

NLP Natural Language Processing

POS Part-of-Speech

RNNs Recurrent Neural Networks

RoBERTa A Robustly Optimized BERT Approach

SVM Support Vector Machine

SemCor Semantic Concordance

Senseval Semantic Evaluation

SpaCy Space Polyglot

word2vec Word to Vector

WSD Word Sense Disambiguation

XML eXtensible Markup Language

Chapter 1

Introduction

The goal of artificial intelligence is to build intelligent systems to mimic human activity, including language understanding [1], learning [2], reasoning [3], and problem-solving [4]. At the heart of this ever-evolving **Artificial Intelligence (AI)** journey lies **Natural Language Processing (NLP)**, a subfield that focuses on bridging the gap between humans and machines in communication, comprehension, and interaction. Yet, language has become an active and challenging research area in natural language processing [5], with its inherent complexities and ambiguities. One of these challenges is **Word Sense Disambiguation (WSD)**, which aims to determine the correct meaning of the ambiguous word within a given context [6].

To understand WSD, consider the word **"bass"** used in two sentences. **Sentence1:** Subwoofer's booming **bass** vibrated the room, immersing everyone in the music. **Sentence2:** Angler cast his line, hoping to catch a largemouth **bass** glinting in the sun. So the word bass in the first sentence refers to **deep musical sound** and **type of fish** in the second. Understanding the intended meaning of a word is simple for humans but it is hard for machines.

WSD plays a crucial role in numerous natural language processing applications, such as machine translation [7, 8], improving question-answering chatbots [9], information retrieval [10], enhancing sentiment analysis [11], and facilitating overall comprehension of natural language.

Over the past years, researchers have explored various methods, since the pioneering days of Warren Weaver's 1940s [12] memo to tackle WSD. Yet, de-

spite decades of exploration through rule-based methods including lesk algorithms [13], corpus-driven approaches, statistical techniques, machine learning algorithms, and even deep learning powerhouses like BERT [14] and Embeddings from Language Models (ELMo). Recently, the utilization of pre-trained models like RoBERTa [15], and GPT have significantly improved word sense disambiguation performance.

However, word sense disambiguation remains a demanding task with many open problems, including polysemy and homonymy, contextual dependency, handling complex sentences, sense granularity [16], fine-grained [16] and coarse-grained [6] sense. Resolving word sense ambiguity has proven to be a challenging task, and it is considered an **AI-complete** problem[17].

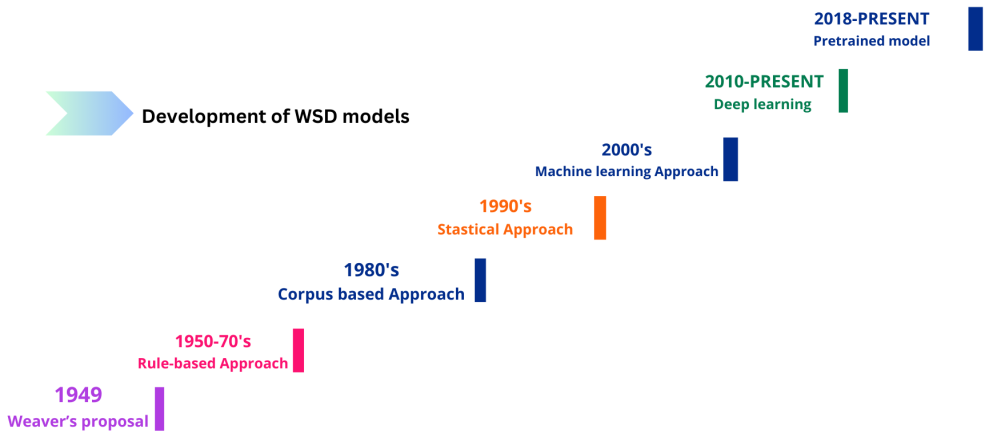


Figure 1.1: Historical model development to word sense disambiguation

Contextualized models are undoubtedly crucial for WSD. Word embedding plays a vital role within this contextualized framework, acting as a bridge that translates the vastness and intricacy of human language into precise points within high-dimensional mathematical spaces. The introduction of word2vec [18] caused a significant stir in lexical semantics, which remains a widely used method. However, it is limited by its lack of contextual sensitivity. GloVe [3] later surpassed

word2vec in word analogy, similarity, and training time. The evolution of NLP led to the development of advanced contextual embeddings like Fasttext [19], ELMO [20], GPT [21], and BERT [14], taking language models to new heights.

Traditional word embeddings, such as Word to Vector (word2vec) and Global Vectors for Word Representation (GloVe), assign a static vector to each word without considering its context, unlike modern large language models like GPT, ELMo, and BERT, which offer a richer contextual understanding. Word embedding serves as more than just a translator between humans and machines; it forms the bedrock of modern natural language processing. Its significance lies in enhancing semantic understanding, capturing the meaning of words in context, and improving model performance across various NLP tasks.

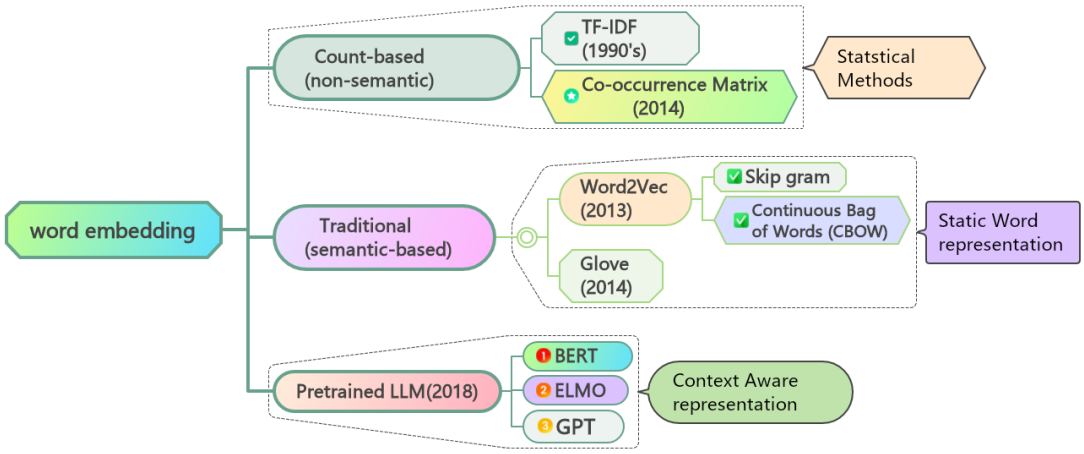


Figure 1.2: Development of word embedding methods

Recently, the utilization of pre-trained models like BERT [14] and its derivative, RoBERTa [15], ELMO, and GPT have significantly improved word sense disambiguation performance. Their capability for extracting context-sensitive semantic information from text seems to make them especially suitable for this task [22]. Attention mechanisms have demonstrably enhanced various NLP tasks, such as machine translation, word sense disambiguation, and sentiment analysis [23]. Their application in WSD has also garnered significant interest [24].

Although recent language models have achieved remarkable success, there has

been a limited amount of in-depth analysis to fully understand their efficacy in lexical semantics, specifically in resolving ambiguity. Despite significant analysis, there is limited research on how neural language models tackle lexical ambiguity.

1.1 Motivation

In the field of artificial intelligence, NLP plays a significant role in facilitating seamless interaction between machines and humans. As our world becomes more interconnected, NLP becomes increasingly important in improving communication systems. However, a challenge arises when words have multiple meanings in different contexts, which is difficult for machines to overcome.

To address this challenge, Word Sense Disambiguation accurately identifies the intended meaning of ambiguous words within a given context [6]. This capability greatly enhances the performance of various NLP tasks such as text summarization [25], machine translation, sentiment analysis [26], question answering [9], information retrieval [27], and language understanding. WSD not only improves the efficiency of these applications but also brings us closer to achieving more natural and effective human-machine communication.

While humans may find Word Sense Disambiguation straightforward, machines struggle to comprehend natural language. This highlights the critical role of WSD in bridging the gap between language understanding and machine processing. Despite the growing interest in this field, WSD requires increased attention due to its significant impact on various NLP applications.

The motivation behind our work is the increasing importance of resolving word sense ambiguity as AI and NLP systems strive to understand and interact with the world. Words often have multiple meanings depending on the context, which poses challenges for accurately interpreting or generating human language in NLP tasks. This challenge is particularly acute in languages like Amharic, where variation carries significant meaning.

By enhancing WSD, especially in languages with complex morphology like Amharic, we aim to unlock the potential of AI and NLP systems to effectively communicate and engage. Additionally, leveraging resources and insights from English language WSD can pave the way for developing cross-lingual WSD models, expanding the reach and impact of these advancements.

1.2 Statment of the Problem

Traditional approaches to word sense disambiguation like Lesk algorithm [13] and Bootstraping [28] often struggle with knowledge bottleneck, then the introduction of unsupervised approach address this problem [29], Neural network [30] then was a paradigm in word sense disambiguation until the introduction of contextual deep learning approach [31]. The introduction of these pre-trained contextualized deep learning methods like BERT [14], ELMO [20], and fastText [19] transform different natural language processing tasks surprisingly to the next level by capturing long-range dependencies and feeding more information to the model.

Despite the advancements in contextual models for natural language processing tasks, word sense disambiguation has not fully benefited from these deep learning models, as they require specific optimization. Using the Attention mechanism for word sense disambiguation has demonstrated remarkable outcomes, particularly in enhancing the contextual understanding of words [32]. Recent work has been conducted to enhance the model's ability to capture contextual features related to the target ambiguous word using different deep learning approaches like Bidirectional Long Short-Term Memory (BiLSTM) since it is a proven method for sequential problems in NLP. While interesting results have been achieved when applying the attention mechanism to word sense disambiguation researchers are unable to distribute more attention towards context tokens, but it is only towards ambiguous words [33].

A major challenge standard self-attention mechanisms face is their limited

consideration of the context. These mechanisms in WSD primarily focus on target words rather than paying attention to whole sentence word context with the target word, disregarding the importance of long-distance relationships and overall meaning. This limitation hinders accurate disambiguation, particularly in complex with intricate structures or context clues. Another concern is that existing models often have the profound meaning of language relying excessively on surface features. Consequently, distinguishing between words with similar meanings becomes challenging, especially for uncommon interpretations dependent on context. As a result, the models generate inaccurate outcomes, restricting their applicability to diverse real-world language.

Despite the positive progress of research to solve word sense disambiguation using the help of contextual models [14, 34, 35, 32, 36, 37], semantic [6], and syntactical [38]ambiguity remain challenging. This research explores the integration of hierarchical attention mechanisms into BiLSTM models with BERT embeddings to improve WSD accuracy, particularly to make the model context-aware and capture important information. By leveraging these techniques and addressing limitations such as capturing multi-scale context, integrating semantic knowledge, and handling rare senses, more robust and accurate WSD models can be developed for various NLP tasks and real-world applications.

Recent NLP research focuses on two main trends: combining different datasets to improve model performance and developing algorithms that pay more attention to the target word within a sentence, while de-emphasizing the importance of other surrounding words (context tokens). Unlike that, this research investigated the use hierarchical attention mechanism towards the target word and much more attention to surrounding context tokens to improve contextual understanding of the model.

Furthermore, this thesis expanded the investigation into the effectiveness of the proposed model in adapting to improve WSD accuracy in low-resource lan-

guages such as Amharic. Amharic word sense disambiguation is more challenging due to the morphological richness [39] of the language, much use of variational alternative words [40], and lack of labeled training data like Wordnet [41]. Overall, this study explores various deep-learning mechanisms with a focus on improved attention techniques to develop a more effective word sense disambiguation model. Our proposed model seeks to address the limitations of current WSD approaches by incorporating hierarchical attention mechanisms, which can better capture contextual nuances and semantic relationships between words.

1.3 Research Question

1. How does incorporating hierarchical attention mechanisms into BiLSTM models with BERT embeddings impact WSD accuracy, particularly in complex sentences or for long-range dependencies?

2. To what extent can the proposed model's enhanced ability to understand contextual information through both local and global attention mechanisms lead to improved word sense disambiguation?

3. Can the proposed model be adapted to improve performance for Amharic word sense disambiguation?

1.4 Objectives

1.4.1 General Objective

To enhance the performance of word sense disambiguation with hierarchical attention and semantic integration using BiLSTM model.

1.4.2 Specific Objectives

- Design and implement hierarchical attention mechanisms, experimenting with multi-level, multi-head for effective context capture within a BiLSTM model.
- Explore the effectiveness of BERT embeddings and attention mechanisms for semantic integration within the BiLSTM architecture, aiming to enhance context representation and address challenges in word sense disambiguation.
- Evaluate the model's performance on English WSD benchmark datasets (Semantic Concordance (SemCor), Semantic Evaluation (Senseval)) and compare it to state-of-the-art systems, including BERT-based models.
- Adapt the proposed model for Amharic word sense disambiguation, and addressing language-specific challenges.
- Analyze the impact of hierarchical attention on complex sentences, and long-range dependencies.

1.5 Significance

As a core to NLP, WSD impacts so many tasks as discussed in section 1.1. This work explores how incorporating hierarchical attention mechanisms into BiLSTM models with BERT embeddings can significantly improve WSD accuracy in complex sentences and for long-range dependencies. This addresses a crucial challenge in WSD and paves the way for more robust natural language understanding in NLP applications. The proposed model's ability to leverage both local and global attention mechanisms opens new avenues for understanding rich contextual information within text. This deeper understanding has the potential to improve the overall performance and effectiveness of WSD models across various tasks. This research investigates the feasibility and effectiveness of transferring

knowledge from English WSD to significantly improve performance in Amharic, a resource-scarce language. This opens doors for bridging the digital divide in NLP capabilities and promoting advancements in diverse linguistic communities.

1.6 Contribution

The main contribution of this paper is presented below. Our research makes several significant contributions to the field of Word Sense Disambiguation. **Firstly**, we propose hierarchical attention mechanisms within BiLSTM models, enhancing the model's ability to capture context at both local and global levels. **Secondly**, we integrate BERT embeddings, exploiting pre-trained contextual information to improve WSD accuracy, especially in complex sentences and for long-range dependencies. Thirdly, we adapt our model to both Amharic and Italian, demonstrating its effectiveness in word sense disambiguation. This highlights the model's potential for application in various low-resource languages. **Additionally**, we address the research gap of the lack of standard benchmark datasets for Amharic WSD by creating and utilizing our own dataset, which can serve as a valuable resource for future research. **Lastly**, our study provides insights into the importance of attention mechanisms and contextual embeddings, offering guidance for future research in WSD for morphologically rich languages. Generally:

- To the best of our knowledge, we are the fist to propose a hierarchical attention mechanism and semantic integration to enhance WSD. The approach close to us uses self-attention [42]. We have achieved 1.8% to 2.9% F1 score for English language over baseline models [43, 42].
- **Effectively Adapted our model to Amharic WSD**: Our BiLSTM model with hierarchical attention and BERT embeddings was applied to word sense disambiguation in Amharic, a low-resource language. We have achieved 78.2% on the test set and 92.4% F1 score on the training set.

- **Robustness in the Italian language**: Our model's effectiveness was further tested in the Italian language, yielding satisfactory results improving 0.5% on Semeval-2013 and 0.7% on Semeval-2015 F1 score over baseline models [44, 45].

- Contributed to addressing the research gap in Amharic WSD by creating a **benchmark dataset**. This dataset will facilitate future research and evaluation of WSD models for Amharic.

- We demonstrate how attention mechanisms and contextual embeddings benefit Word Sense Disambiguation tasks, providing insights for **future research** with our proposed BiLSTM model.

1.7 Scope

This research aims to build a contextual model, specifically a hierarchical attention-based BiLSTM neural network with BERT embeddings, to address semantic ambiguity in both English and Amharic languages. The scope encompasses utilizing the SemCor dataset [46] for nouns, verbs, and adjectives to train and evaluate the model's ability to disambiguate words based on syntactic and semantic contexts, particularly in sentences with complex structures or long-range dependencies. Additionally, standard testing datasets like semeval2007 [47], semeval2013 [48], and semeval2015 [49] are used to assess the model's performance on various WSD tasks. For the Amharic language, due to the lack of publicly available benchmark datasets, this research utilizes a custom dataset specifically prepared for this task. The dataset encompasses 10k sentences containing ambiguous words collected from different Amharic news outlets, dictionaries, and religious books, and types of ambiguity covered. The model's performance in Amharic was evaluated on this custom dataset. Furthermore, our research doesn't include a sentence that is too short and requires more context, which falls outside the scope of our

current objectives.

1.8 Thesis Structure

The thesis is Organized into 5 chapters. chapter 1 serves as an introduction, presenting the research problem, objectives, and methodology. chapter 2 discusses the foundations of deep learning, examining relevant concepts and techniques, with a focus on Transformers and related works. chapter 3 discusses the proposed methodology for enhancing WSD, detailing the BiLSTM model with BERT embedding and addressing three identified weaknesses. In chapter 4, the experimentation and findings are presented, covering the setup, models, algorithms, analysis of experimental resuls, and findings. Finally, chapter 5 concludes the study by concluding remarks and providing recommendations for future research directions.

Chapter 2

Litrature Review

Overview

The beginning of this chapter explains the concept of word sense disambiguation section 2.1. Following this, discusses the historical perspective of the development of word sense disambiguation starting from the first machine translation memo to advanced research using deep learning. After that, section 2.2 presents a brief description of earlier methods developed for word sense disambiguation using the traditional approach like the Lesk algorithm, corpus base, statistical approach, and machine learning models. section 2.3 discusses the recent advancements in deep learning models for Word Sense Disambiguation, including a brief analysis.

Then subsection 2.3.3 discusses the role of attention mechanisms in NLP and different types of attention mechanisms, specifically utilizing attention mechanisms for improved contextual information is discussed. Subsequently, subsection 2.3.5 examined techniques for semantic integration using different embedding techniques including traditional word embedding (word2vec, glove), and advanced embedding like BERT. And presenting the way to integrate semantic understanding into neural network models for improved language comprehension. Lastly, section 2.5 examines state-of-the-art papers and related works, identifying existing research that provides the basis for the proposed method.

In this chapter, We conducted a comparison among WSD models, identified various instances of word sense ambiguities, emphasised areas that have not been thoroughly addressed, and examined the attention mechanisms employed in

transformer models along with their respective strengths and weaknesses.

2.1 Word Sense Disambiguation

Word sense disambiguation is the task of natural language processing which aims to identify the proper sense of ambiguous words within a given context[6]. In language, there are ambiguous words that have multiple meanings, known as Homonyms. For example, the word **bank** can refer to a financial institution or the side of a river. Some common types of word sense ambiguity are Polysemy, Homonymy, Synonymy, Antonymy, Hyponymy/Hypernymy, Meronymy/Holonymy, Semantic Ambiguity, and Syntactic Ambiguity.

SemCor [41] is a well-known training dataset for word sense disambiguation tasks it is based on wordnet synsets. For evaluation purpose there are dataset like senseval-2 [50], senseval-3 [51], senseval-2007 [47], senseval-2013 [48], senseval-15 [49].

Table 2.1: Dataset Statistics

Dataset	Noun	Verb	Adj	Adv	Total
SemCor	87002	88334	31753	18947	226036
SE2	1066	517	445	254	2282
SE3	900	588	350	12	1850
SE07	159	296	0	0	455
SE13	1644	0	0	0	1644
SE15	531	251	160	80	1022

Wordnet [41] is a huge data source for WSD which is prepared manually by lexicographers into organized information that contains structured synsets linked with various semantic relations as shown in the following table.

Table 2.2: Example Words from WordNet

Sense	Context	Definition
Bank1	She deposited her paycheck at the **bank**.	A financial institution
Bank2	We had a picnic by the river **bank**.	Side of a river
Mouse1	Be careful, there's a **mouse** in the kitchen.	A small rodent animal
Mouse2	Click the left **mouse** button to select an item.	Computer input device used to move a cursor

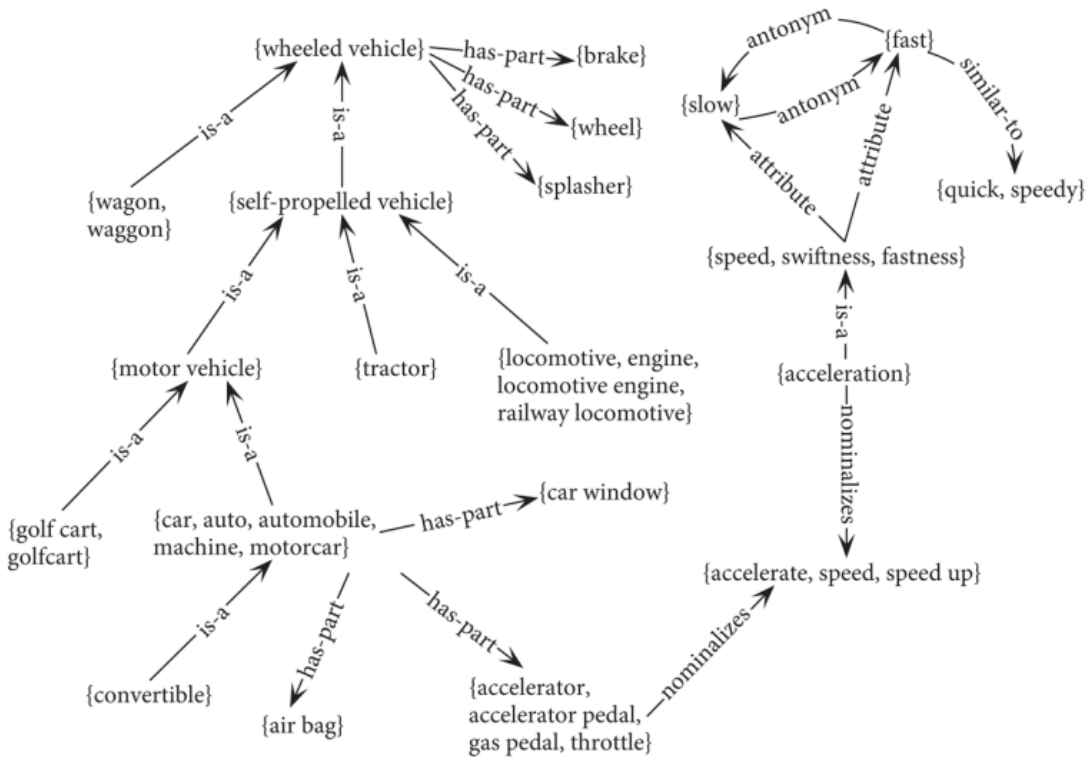


Figure 2.1: Wordnet view in graph. Figure source [52]

As the above figure shows wordnet prepared along with Synset information, we can use a sense or superset of sense if we want a coarser-gain set.

2.1.1 Fundamental Tasks in WSD

The following list of essential steps must be taken to complete the process of word sense disambiguation.

Word Sense Identification: The first crucial step and central focus of WSD

is determining a word's sense in a particular context. Unlike many other classification problems, word senses are complicated and challenging to discretize into a small, discrete set of entries, each expressing a distinct meaning [53]. **Use of external knowledge resources:** Word sense disambiguation heavily depends on external resources. One of the most important resources is using sense inventories, like WordNet and BabelNet, which systematically organize words into synsets. To effectively traverse the complex network of language, sensee inventories are essential in offering an accurate representation of word senses [54]. Both structured and unstructured sources can be used as knowledge sources. Ontologies, thesauri, and machine-readable dictionaries (MRDs) are a few examples of structured sources. On the other hand, corpora and collocation resources are unstructured sources.

Sense Annotation: Context words are manually labeled with the most suitable meanings. This entails a series of preliminary tasks such as chunking, parsing, normalization, lemmatization, tokenization and part-of-speech tagging. Then global features, syntactic features, semantic features, and local features are considered.**Choosing Appropriate Approach:**Different approaches include supervised approach, knowledge-based approach, unsupervised approach and deep learning approach in word sense disambiguation. Thus it is important to choose an appropriate approach according to our data and specific task.

The task of WSD is taking a sentence as a puzzle of words (w1, ..., wm). Some of these words, let's call them targets (t1, ..., tk), have multiple possible meanings. The goal of WSD is to figure out the most appropriate meaning (like picking the right puzzle piece) for each target word based on the context of the sentence. This context helps us narrow down the options, typically from a predefined dictionary or reference like WordNet [41], and Babelnet [55] where each word has multiple entries representing its different meanings.

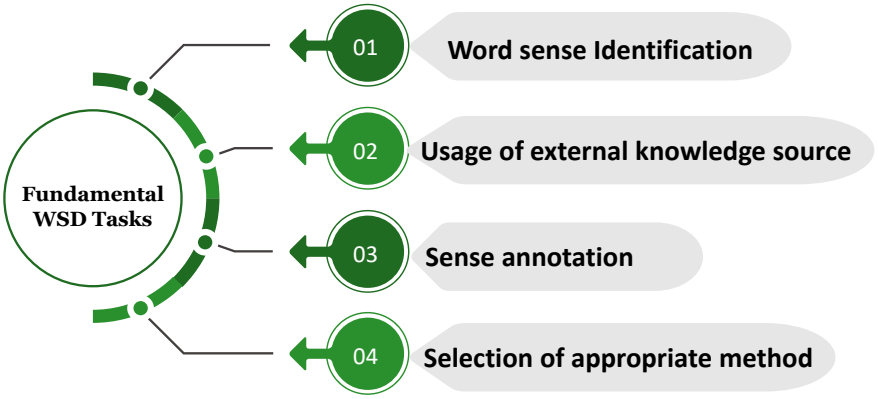


Figure 2.2: Task Summary in WSD

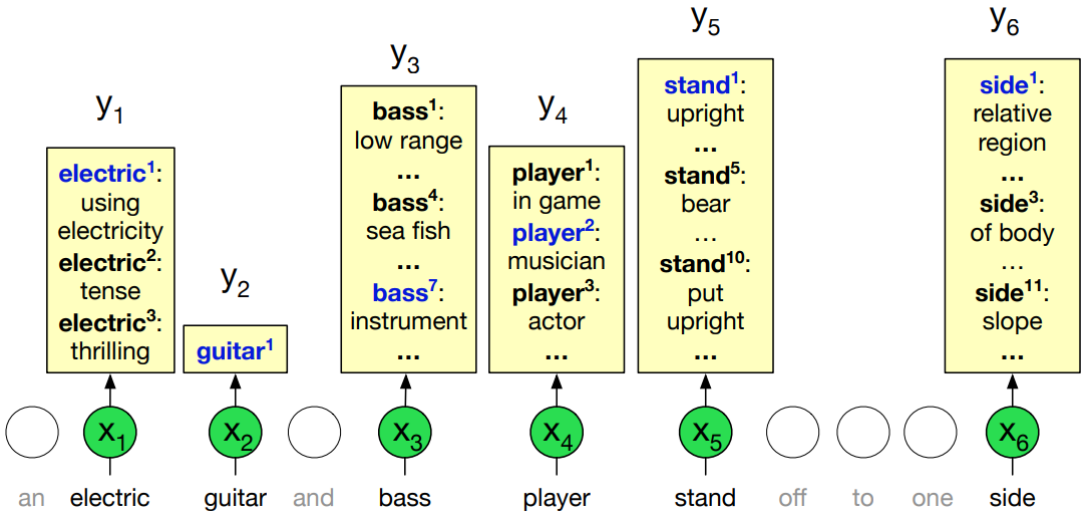


Figure 2.3: The all-words WSD task, mapping from input words (x) to WordNet senses (y). Only nouns, verbs, adjectives, and adverbs are mapped, and note that some words (like guitar in the example) only have one sense in WordNet. Figure taken from paper [56]

2.2 Traditional Approaches to WSD

WSD has been a topic of interest in NLP since the early days of computational linguistics. The development of WSD techniques started with the earliest attempts in the work of Warren Waver in 1949 [12], who proposed using a dictionary and a set of rules to disambiguate word senses in machine translation systems.

In the 1970s and 1980s, researchers shifted to statistical and probabilistic methods. A significant breakthrough came with Michael Lesk's development of the Lesk algorithm in 1986 [13]. Moving into the 1990s, interest surged in machine learning for WSD, with researchers employing decision trees, support vector machines, and neural networks. One pivotal contribution was Yarowsky's Senseval-1 paper in 1995 [28], which introduced the Senseval datasets for evaluating WSD systems. In the 2000s, there was a shift to using large lexical resources like WordNet and extensive text corpora such as the Brown Corpus and the Penn Treebank. Let's see the traditional approach to word sense disambiguation below.

2.2.1 Knowledge/ Dictionary Based Approach

Knowledge-based approaches rely on the use of lexical resources, such as dictionaries and thesauri, to disambiguate word senses [57]. The first work that serves as a foundation for today's modern algorithms was articulated by Warren Waver [58] in the context of machine translation. Following this work, the use of thesaurus for disambiguation was introduced [59], supervised training of Bayesian models [60], and the use of clustering in word sense analysis [61].

The most well-known algorithm for WSD called the Lesk algorithm [13], which considers overlapping word senses in context, measures of semantic similarity computed over semantic networks, heuristic methods [62], and senses acquired automatically or semi-automatically.

Lesk Algorithm: Introduced by Michael Lesk in 1986 on the idea of **word over-lap**. The algorithm examines word pairs to determine the meaning of a phrase. It compares the definitions of surrounding words against possible senses of the target word. This allows it to select the best sense in context.

Algorithm 1 SIMPLIFIED LESK Algorithm

```
function SIMPLIFIED_LESK(word, sentence)
    best-sense ← most frequent sense for word
    max-overlap ← 0
    context ← set of words in sentence
    for each sense in senses of word do
        signature ← set of words in the gloss and examples of sense
        overlap ← COMPUTEOVERLAP(signature, context)
        if overlap > max-overlap then
            max-overlap ← overlap
            best-sense ← sense
        end if
    end for
    return (best-sense)
end function
```

Table 2.3: Base line paper in knowledge base approach with their evaluation

	Evaluation (F1-score for each gold standard datasets)						
Model	Senseval-2	Senseval-3	Senseval-2007	Senseval-2013	Senseval-2015	All	Year
Lesk_ext	50.6	44.5	32.0	53.6	51.0	48.7	2003 [63]
Lesk_ext+emb	63.0	63.7	56.7	66.2	64.6	63.7	2014 [64]
UKB	68.8	66.1	53.0	68.8	70.3	57.5	2014 [65]
UKB_gloss	64.2	54.8	40.0	64.5	64.5	57.5	2014 [65]
WN 1st sense baseline	66.8	66.2	55.2	63.0	67.8	65.2	2017 [66]
Babelfy	67.0	63.5	51.6	66.4	65.5	70.3	2014 [67]
WSD-TM	69.0	66.9	55.6	69.6	65.3	66.9	2018 [56]
kEF	56.9	72.3	69.6	66.1	68.4	68.0	2020 [68]
SCSMM	68.9	67.6	57.1	63.5	69.5	66.7	2022 [69]

2.2.2 Graph-Based and Semantic Network Approaches

Quillian [70] introduced a graph-based method for language processing, representing word definitions as networks of interconnected word nodes linked by syntactic and semantic relationships. This approach aimed to disambiguate word senses by identifying the shortest path between senses in the graph. In 1973 Simmons [71] further contributed to this field with another influential semantic network approach.

Wilks [72] proposed Preference Semantics, which was one of the earliest non-discrete models in this area. Riesbeck suggested understanding systems that

focused on modeling detailed procedural information for each word. Hirst's ABSITY system [73] was a notable advancement, utilizing marker passing based on semantic networks.

In parallel, early neural network approaches to WSD, then known as 'connectionist' approaches, also emerged. These approaches, exemplified by Cottrell and Kawamoto [74], relied on small lexicons with handcrafted representations.

2.2.3 Machine learning and statistical approach

Supervised Approach

By utilizing a collection of words labeled with their meanings in specific contexts, the supervised approach to word sense disambiguation teaches a model to label words in new texts. While this method harnesses machine learning techniques to effectively clarify meanings and grasp ambiguous relations, it relies on having labeled training data, which can be both expensive and time-intensive to create. Commonly used algorithms in supervised learning include XGBoost, AdaBoost, Neural Networks, Naive Bayes, Support Vector Machines, and Decision Trees [75].

The earliest supervised disambiguation methods involved using decision trees by [76]. A decision tree is similar to a flow chart, according to Jumi Sarmah[77], with internal nodes standing in for tests, branches for test findings, and leaves for sensory levels.

Then following this there was highly effective and computationally efficient on small datasets called **Naive Bayes** algorism [78]. The Naive Bayes algorithm shines with its flexibility (both numbers and words) making it suitable for linguistic context, but its simplistic assumption, makes it perform poorly with strongly correlated features and struggle to recognize complex language relationships. Matimathecally, the naive Bayes algorithm for word sense disambiguation is worked

by choosing the best sense $\hat{s}$ out of the set of possible senses S for a feature vector $\tilde{f}$ amounts to choosing the most probable sense given that vector.

$$\hat{s} = \arg\max_{s \in S} P(s|\tilde{f})$$

Compared to other supervised methods, it has been shown that Support Vector Machine (SVM) yields the best results in WSD[79]. Nevertheless, SVM needs large annotated datasets for training and can be computationally expensive. **Neural network** models provide exceptional performance in several natural language processing problems by deriving intricate representations from the data. Furthermore, as neural networks are multilingual, they can adapt to different linguistic contexts [30].

Table 2.4: Comparison of Supervised and Knowledge-base WSD Approaches

Approach	**Description**	**Strengths**
MFS	Selects the sense with the highest frequency in the training data.	Simple and efficient.
Leskext_+emb [13]	Leverages word similarity in a distributional semantic space to assess alignment between glosses and contextual usage.	Augments gloss information through semantic relationships.
UKB [65]	Utilizes the UKB knowledge base to provide context-aware disambiguation.	Overcomes limitations of MFS by incorporating external knowledge.
Babelfy [67]	Employs the semantic network structure of BabelNet for WSD and Entity Linking.	Similar to UKB, overcomes limitations of MFS by incorporating external knowledge.
Context2Vec [80]	Learns distributed representations of words and contexts using a supervised approach.	Addresses limitations of knowledge-based approaches.
IMS [81]	Employs a linear SVM classifier with features like POS tags and local collocations.	Simple and efficient.
IMS+emb [82]	Extends IMS by incorporating word embeddings.	Improves performance over IMS.
Seq2Seq [43]	Explores different variants of the Seq2Seq model for WSD.	Demonstrates potential for neural network approaches.

We introduced foundational work in Table 2.4 for word sense disambiguation, which serves as a basis for our research as well as many others. Most papers, except **Seq2Seq**, use the Most Frequent Sense (MFS) baseline. However, these approaches are limited in their generalizability and ability to handle unseen words.

Table 2.5: Baseline papers for the Supervised approach with their evaluation

	Evaluation (F1-score for each gold standard dataset)					
Model	Senseval-2	Senseval-3	Senseval-2007	Senseval-2013	Senseval-2015	Year
ELMo	71.6	69.6	62.2	66.2	71.3	2018 [83]
Bi-LSTM_att+LEX+POS	66.9	69.1	64.8	71.5	72.0	2017 [66]
BiLSTM_att+LEX	72.0	69.4	63.7	66.4	72.4	2017 [42]
GAS	72.0	70.0	-	66.7	71.6	2018 [84]
BERT	73.8	71.6	63.3	69.2	74.4	2019 [36]
EWISER	78.9	78.4	71.0	78.9	79.3	2020 [44]
SemCor_WNGC+hypernyms	79.7	77.8	73.4	78.7	82.6	2019 [85]
EWISER+WNGC	80.8	79.0	75.2	80.7	81.8	2020 [44]
ESR	81.3	79.9	77.0	81.5	84.1	2021 [86]
ESR+WNGC	82.5	80.2	78.5	82.3	85.3	2021 [86]
ConSeC+WNGC	82.7	81.0	78.5	85.2	87.5	2021 [87]

Unsupervised Approach

Unsupervised methods offer a way to bypass the knowledge acquisition bottleneck[29]. This method operates without the need for labeled data and instead utilizes clustering or topic modeling methods[88]. The primary goal is to group words with similar contexts, assuming they share similar senses. Commonly used unsupervised methods include context clustering, word clustering, and co-occurrence graphs[89]. A bootstrapping is another supervised model presented by Yarowsky in 1995[90]. It starts with a few sense-labeled examples and gradually assigns appropriate senses to cases that are not yet labeled.

2.3 Advancements in Deep Learning for WSD

Deep learning methods have significantly improved in the last several years in several natural language processing applications, such as word sense disambiguation. By automatically learning feature representations and utilizing contextual data, deep learning techniques have shown state-of-the-art performance in a variety of NLP tasks[91], including WSD. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and transformer models such as BERT and GPT show promise in capturing complex word relations and representations[31].

2.3.1 Multilayer Perceptron

Only since the start of the decade 2010 has there been a greater focus on deep learning experiments and their application to the WSD problem, namely with Senseval 3 tasks and Senseval [92]. The research proposed by Nguyen et al. [93] uses a special type of neural network (multilayer perceptron) to automatically determine the correct meaning of words in a sentence. Unlike usual neural networks, this one pushes borderline words away from their current prediction instead of just adjusting weights.

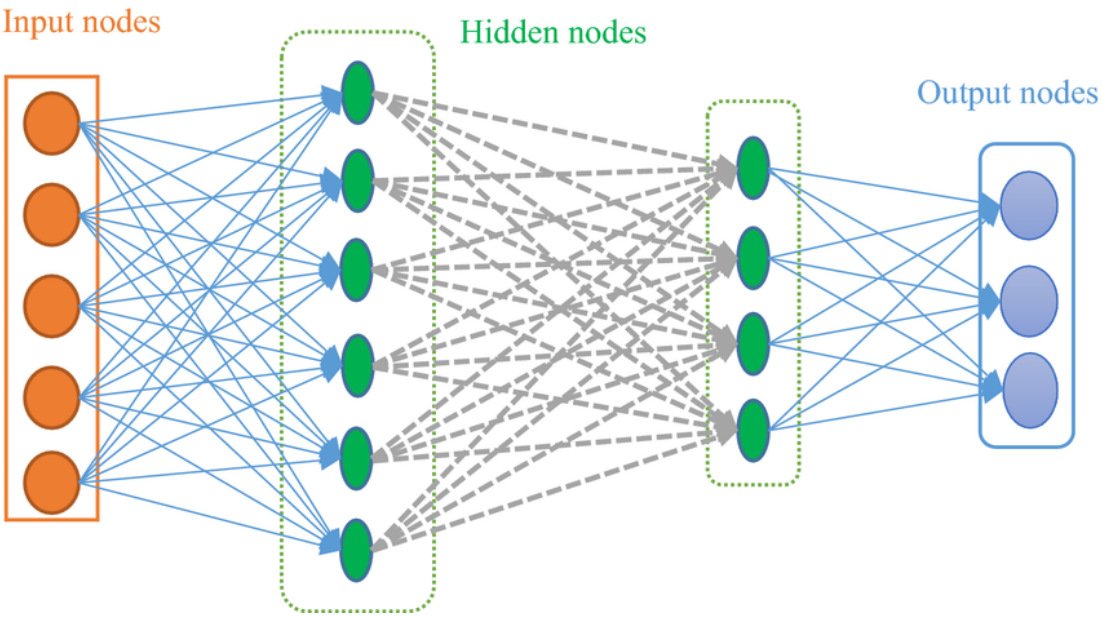


Figure 2.4: Multilayer feed-forward NN. Figure taken from [94]

2.3.2 Recurrent Neural Networks for word sense disambiguation

Recurrent Neural Networks (RNNs) developed as a viable tool for modeling language. RNNs were historically challenging to train due to the recurring issue of the exploding or vanishing gradients problem, wherein the backpropagated error gradients become excessively large or small, hindering their memory capabilities, particularly evident in long-time series data [95].

In contrast to standard neural networks, recurrent neural networks process individual words as inputs rather than entire samples, allowing them to adapt to varying sentence lengths, flexibility unattainable in fixed-structure standard neural networks.

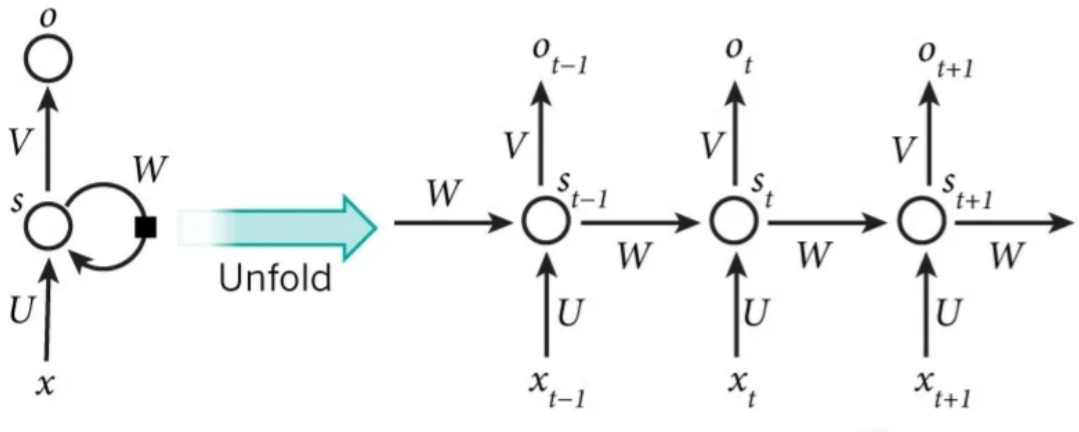


Figure 2.5: RNN architecture for WSD.Source [96]

With ($T_x = T_y$), or number of inputs = number of outputs, the architecture shown above figure 2.5 is also known as a many-to-many architecture. A structure like this is quite helpful for sequence modeling.

Long Short-Term Memory (LSTM) [97] and Gated Recurrent Unit (GRU) [98] is then designed to overcome *vanishing gradient* problem of RNN. The GRU includes an extra memory unit known as an update or reset gate, alongside the standard sigmoid and softmax units for processing. The additional unit uses tanh activation, which allows for both amplifying and reducing output values, and its output is used to update the memory cell value in combination with the activation input.

GRUs, similar to LSTMs, utilize gates to regulate information flow and are a relatively newer variant, offering some enhancements over LSTMs with a simpler architecture. Mathematically, LSTM is defined as the following: it uses three gates: forget gate (f), input gate (i), and output gate (o). Each gate takes the previous hidden state ($h^{(t-1)}$) and current input ($x^{(t)}$) as input and outputs a value between 0 and 1 through a sigmoid function.

$$\text{Forget gate: } f^{(t)} = \sigma(W_f \cdot [h^{(t-1)}, x^{(t)}] + b_f) \tag{2.1}$$

$$\text{Input gate: } i^{(t)} = \sigma(W_i \cdot [h^{(t-1)}, x^{(t)}] + b_i) \tag{2.2}$$

$$\text{Output gate: } o^{(t)} = \sigma(W_o \cdot [h^{(t-1)}, x^{(t)}] + b_o) \tag{2.3}$$

$$\text{Update state: } c^{(t)} = f^{(t)} \cdot c^{(t-1)} + i^{(t)} \cdot \tanh(W_c \cdot [h^{(t-1)}, x^{(t)}] + b_c) \tag{2.4}$$

The key difference between LSTMs and GRUs lies in their gate structures and mechanisms for updating hidden states and cell states. LSTMs utilize three gates and separate mechanisms for updating cell states, while GRUs employ only two gates and streamline the update process with combined mechanisms as expressed mathematically as follow:

$$\text{Update gate: } z^{(t)} = \sigma(W_z \cdot [h^{(t-1)}, x^{(t)}] + b_z) \tag{2.5}$$

$$\text{Reset gate: } r^{(t)} = \sigma(W_r \cdot [h^{(t-1)}, x^{(t)}] + b_r) \tag{2.6}$$

$$\text{Updated cell State: } c^{(t)} = (1 - z^{(t)}) \cdot c^{(t-1)} + z^{(t)} \cdot \tanh(W_c \cdot [r^{(t)} \cdot h^{(t-1)}, x^{(t)}] + b_c) \tag{2.7}$$

The above equations describe the operation of a Gated Recurrent Unit (GRU), a type of recurrent neural network (RNN). The update gate $z^{(t)}$ controls how much of the previous cell state $c^{(t-1)}$ should be retained, while the reset gate $r^{(t)}$ determines how much of the previous hidden state $h^{(t-1)}$ should be forgotten. The updated cell state $c^{(t)}$ is a combination of the previous cell state and a new candidate state, with the update gate controlling the balance between the two.

For the problem of long-term dependency, the liner LSTM was modified as BiLSM by adding one extra layer on LSTM. BiLSTM stands out for their ability to connect past information to the current task, making them ideal for tasks

like text analysis, where previous words can help determine the meaning of the current word in a sentence.

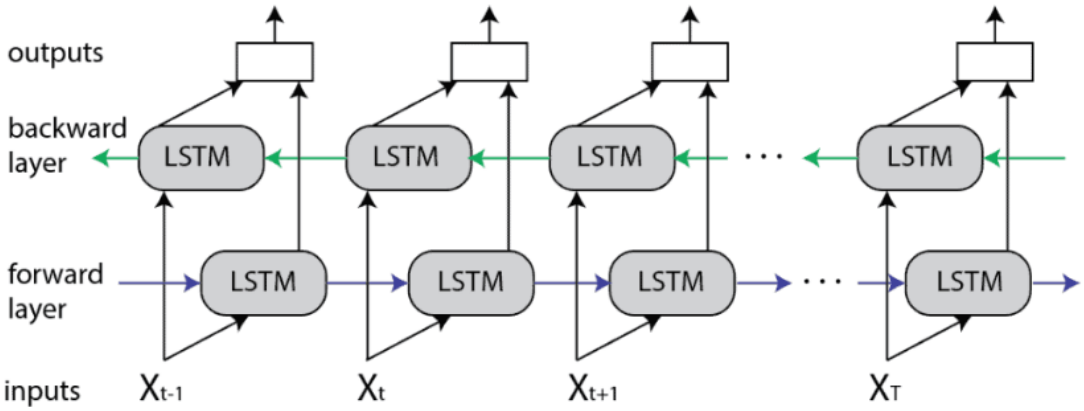


Figure 2.6: BiLSTM architecture.Source [99]

2.3.3 Attention Mechanisms

Attention mechanisms in neural networks are inspired by human visual attention, which allows us to focus on specific parts of an image or scene. In NLP, attention mechanisms enable the model to focus on relevant parts of the input data, such as specific words in a sentence, when making predictions.

The key idea behind attention mechanisms is to compute a set of attention weights that indicate how much focus the model should place on each input element when generating an output. These attention weights are computed dynamically based on the similarity between the current state of the model and each input element. Elements that are more relevant to the current context receive higher attention weights, while less relevant elements receive lower weights.

Since the simplicity of the encoder-decoder model with the final hidden state feed to the decoder contains a single representation, the model is unable to generalize everything about sequential context which is a big problem for this model. The attention mechanism solves this by providing all information from all hidden states. The most common attention mechanisms are Self-attention

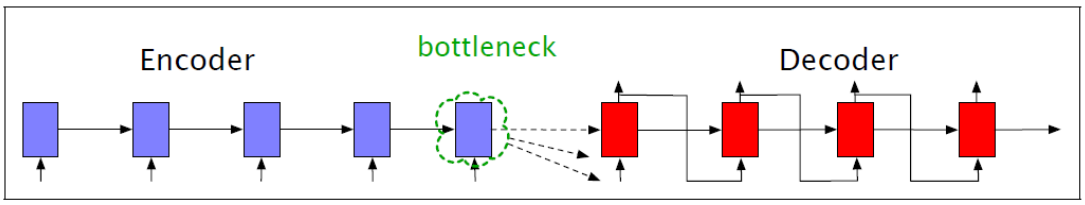


Figure 2.7: Encoder decoder architecture. Source [100] Requiring the context c to be only the encoder's final hidden state means that all the information from the source sentence must flow through this single representational bottleneck.

and Multi-head Attention. Given a sequence of input vectors $X = [x_1, x_2, ..., x_n]$, self-attention computes a set of attention weights $A = [a_1, a_2, ..., a_n]$ as follows:

$$a_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) \cdot V_i \tag{2.8}$$

In a Transformer model, each input vector x_i is transformed into query, key, and value vectors Q_i, K_i, and V_i using learned weight matrices W^Q, W^K, and W^V. The dimensionality of the key vectors is denoted by d_k. The softmax function is then applied to normalize the dot products of the query Q_i with all keys K_j across the sequence, which is a crucial step in the self-attention mechanism.

In multi-head attention, the self-attention mechanism is applied multiple times in parallel, each with its own set of weight matrices W_i^Q, W_i^K, and W_i^V. The outputs of these parallel self-attention mechanisms are concatenated and linearly transformed to produce the final output. The formula for multi-head attention is:

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h)W^O \tag{2.9}$$

2.3.4 Transformer Models

The Transformer is a type of neural network architecture introduced in the paper "Attention is All You Need" by Vaswani et al. [23]. It has been widely used in NLP tasks due to its ability to handle long-range dependencies. The key components of a Transformer model include self-attention mechanisms, multi-

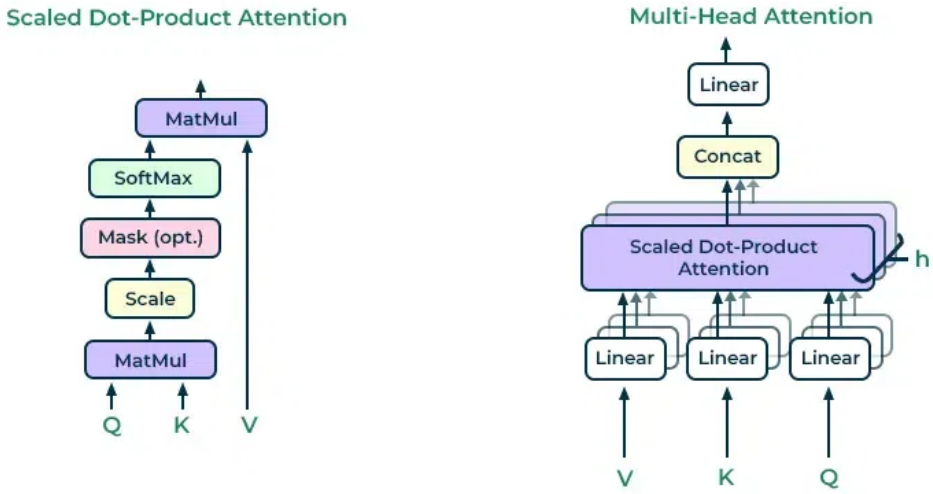


Figure 2.8: Self-Attention layer.Source [23]

head attention, position-wise feedforward networks, and layer normalization.

The **self-attention mechanism** allows the model to weigh the importance of different words in the input sequence when predicting the next word in the sequence. This mechanism enables the Transformer to capture dependencies between words that are far apart in the input sequence. **Multi-head attention** extends the self-attention mechanism by allowing the model to jointly attend to information from different representation subspaces at different positions. This enables the model to focus on different parts of the input sequence simultaneously.

Position-wise feedforward networks are used to transform the output of the attention mechanisms into a form that can be used for the next layer of the network. These networks consist of fully connected layers with a ReLU activation function. **Layer normalization** is applied before each sub-layer of the Transformer model, including the self-attention and feedforward layers. It helps stabilize the training process and speeds up convergence by normalizing the inputs to have zero mean and unit variance.

In recent years, Transformer models have undergone remarkable advancements, propelling the field of NLP to new heights. One of the most notable breakthroughs is the introduction of Bidirectional Encoder Representations from

Transformers (BERT) by Devlin et al. 2018 [14]. Building on BERT's success, further research has concentrated on improving and expanding its functionalities. There are now other variants that offer distinct advances in model efficiency, scalability, and performance, including A Robustly Optimized BERT Approach (RoBERTa) [15], DistilBERT, and ALBERT.

OpenAI's invention of the Generative Pre-trained Transformer (GPT) [21] series is another significant breakthrough. Improving language in both coherent and contextually relevant is shown by GPT models, such as GPT-2 and GPT-3. More specifically, GPT-3 has proven to be capable on a range of language tasks that match human performance, with its 175 billion parameter scale.

Overall, the Transformer model has achieved state-of-the-art results in various natural language processing tasks, including machine translation, text summarization, WSD, and language modeling.

The BERT model is pre-trained using two new unsupervised prediction tasks and a large corpus. Specifically, the next sentence prediction task and the masked language model are employed in pre-training [34]. The use of BERT most of the time by fine-tuning it is applied successfully for word sense disambiguation achieving state-of-the-art performance [34]. Bert can be applied for word sense disambiguation as contextual representation, Relevance ranking, Sequence-pair ranking, and Context-gloss pairs.

2.3.5 Contexual Embedding models

Contextual embedding methods have revolutionized natural language processing by enabling models to understand the meaning of words in context. These methods, including word embeddings and contextualized word representations.

Traditional word embeddings, such as Word2Vec [18] and GloVe [101], represent words as fixed vectors in a continuous space. Despite capturing semantic similarities between words, it ignores contextual variations in meaning. This lim-

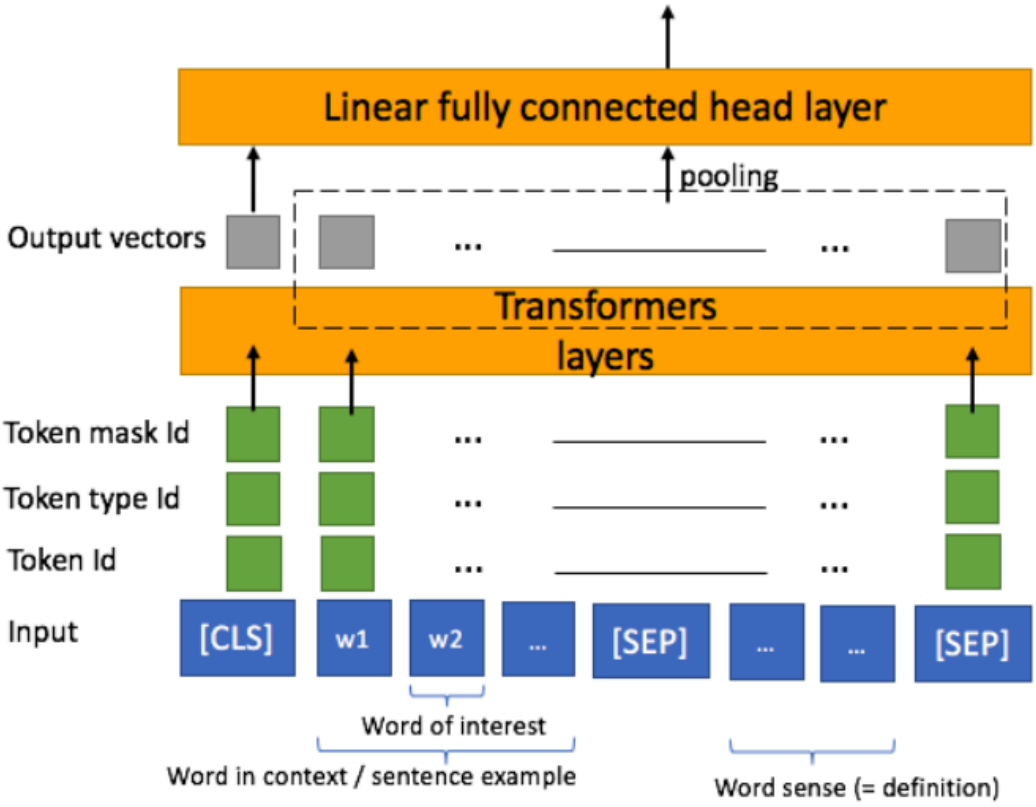


Figure 2.9: Tranformer model for word sense disambiguation.Source: [22]

itation becomes a challenge in WSD, where the meaning of a word can change based on its context.

Contextualized word representations address this limitation by considering the context in which a word appears. ELMo [20], for example, uses a bidirectional LSTM to generate word embeddings that capture the meaning of a word based on its surrounding words. This allows ELMo to capture complex contextual information and improve performance on WSD tasks.

BERT [14] takes contextualized embeddings to the next level with its transformer architecture, which enables bidirectional learning of word representations. This means that BERT can leverage both preceding and following words to understand the context and meaning of a word. As a result, BERT has achieved state-of-the-art performance on various NLP tasks, including WSD.

Contextual embedding methods, despite having advantages, still have some

limitations. These are that they need huge amounts of data for training and costly computational requirements which can be a problem in settings with scarcity of resources. They may also fail at new words or words with rare senses because their dictionaries are not comprehensive enough for such purposes.

In conclusion, contextual embedding methods have significant advantages for WSD by capturing ambiguous contextual information. However, their limitations should be considered when applying them in practical settings, and researchers are actively working on addressing these challenges to further improve the effectiveness of contextual embedding methods.

2.4 WSD Across Different Languages

2.4.1 WSD in English Language

In the English language, word sense disambiguation is a well-researched NLP task. Much of the history of WSD has been determined by the availability of manually created lexical resources in English, including SemCor [105] and wordnet. Then the introduction of BabelNet [106] was a massive multilingual semantic network, created by automatically integrating WordNet, Wikipedia, and other resources.

In 1995, Yarowsky [107] achieved over 95% accuracy in Word Sense Disambiguation using a semi-supervised approach for 12 words. Bootstrapping was employed to train a high-precision word sense disambiguation classifier. According to Kilgarriff and Palmer (2000) [108], the Senseval-1 evaluator achieved a 77% accuracy rate in the English lexical sample task in 1997, while human performance, determined by inter tagger agreement, reached 80%. Stevenson and Wilks employed Part-of-Speech data on all word WSD in 2001 and got an accuracy rating of 94.7% [109]. Due to the utilization of WordNet, Senseval-2 produced a lower score (Edmonds and Cotton 2001) [110], prompting the necessity for further development in Senseval-3. This development ultimately resulted in its top systems

Table 2.6: Summary of WSD Approaches

Approach	Advantages	Disadvantages
Knowledge-based	• Can provide explicit sense definitions and relationships [102] • Interpretable and explainable	• Limited data coverage. • Can't handle new words • Can't capture contextual features
Supervised	• High accuracy with sufficient data • Can capture complex context • Flexibility to use various features	• Requires large annotated data • Difficulty in handling new words
Unsupervised	• Doesn't require labeled data [88] • Ability to discover sense clusters • Can handle new word senses	• Less accurate compared to supervised methods • Limited Interpretability and explainability
Semi-Supervised	• Leverage large amount of unlabeled data • Reduces the need for manual annotation	• Performance is highly dependent on the quality of data
Deep Learning	• Ability to automatically learn complex representations [31] • Can capture long-range contextual dependencies • Can handle multiple languages and contexts effectively	• Requires large amounts of data for effective training • Model interpretability can be challenging • Computationally expensive for training and inference

Table 2.7: State of the art Deep learning methods for word sense disambiguation.

	F1-score			**Accuracy**		
Model	Senseval-2	Senseval-3	All	General-purpose	Domain-specific	Year
BLSTM	66.9	73.4	-	-	-	2019 [103]
FastText	-	-	53.7	56.2	50.6	2020 [104]
BERT-base	-	-	75.3	73.3	77.9	2020 [104]
Transformer	-	-	77.8	75.2	81.0	2020

Table 2.8: Comparisons of WSD approaches

	Approaches			
Criteria	knowledge-based	Supervised	Unsupervised	Deep learning
Data Needed	annotated	Labeled data	Corpus	mixed
Efficiency	High	Moderate	High	Moderate
Scalability	Low	Low	High	High

Note: The "Mixed" category in the "Data Needed" column indicates a combination of annotated and unlabeled data. Scalability is labeled based on the need for annotated or labeled data during model training.

performing at human levels on the English lexical sample test [108].

State-of-the-art English WSD methods utilize deep learning and transformer-based architectures, like BERT (Bidirectional Encoder Representations Transformer) and its variants, which have shown exceptional performance in various NLP tasks, including WSD [34]. BERT uses extensive pretraining on a large corpus to learn contextualized word representations. These pre-trained models can be fine-tuned with specific WSD data to enhance disambiguation accuracy.

2.4.2 WSD in Chinese Language

Whatever the language, there is an ambiguous word in it, For example, the Chinese word "zu" has two common meanings, which are "si wang" and "shi bing" [115]. Word sense disambiguation in Chinese has special challenges due to the language's grammatical uniqueness. The inherent ambiguity resulting from the nature of Chinese characters is one significant difficulty. Furthermore, Chinese language lacks clear word borders, adding complexity that makes strong word segmentation necessary to accurately identify individual words for disambigua-

Table 2.9: Summary of papers in English language WSD

Paper	Technique	Dataset	Language
Lesk (1986)[13]	Knowledge based	OALD	English
D.Yarowsky (1995)[90]	unsupervised learning	Own	English
Kilgarriff et al.(2003)[16]	Supervised approach	Senseval	English
Agirre et al.(2006)[65]	unsupervised approach	Own	English
R. Navigli et al.(2012)[6]	Automatic construction	Wordnet	Multilingual
R. Chain et al.(2013)[88]	unsupervised methods	Own	English
A R. Pal et al.(2015)[110]	Hybrid approach	Own	English
D.S. Chaplot et al. (2018)[56]	Knowledge based	Wordnet	English
J. Devlin et al.(2018)[14]	BERT	Corpus	English
F. Luo et al.(2018)[111]	Co-attantion mechanism	Own	English
L Huanget al.(2019)[34]	BERT based models	Own	English
C. Hadiwinoto et al.(2019)[36]	Pretrained models	Multiple	English
Y. Liu et al.(2019)[15]	RoBERTa	Corpus	English
Y. Luan et al.(2020)[112]	Novel	Babelnet	Multilingual
Z. Lan et al.(2019)[113]	ALBERT	Corpus	English
H.Kanget al.(2023)[114]	Pre-trained model	XL-WSD	Multilingual
D. Loureiro (2023)[35]	Deep learning	Wordnet	English

tion. Furthermore, because a word's meaning can change depending on its environment, contextual changes and syntactic flexibility in Chinese phrases present difficulties.

Researchers have explored various techniques in Chinese WSD, ranging from neural networks to domain adaptation, from word embeddings to cross-language transfer learning. One type of Chinese knowledge resource is typically used in the conventional graph-based Chinese Word sense disambiguation method, which is extremely affected by the knowledge bottleneck issue [116]. Compared with knowledge resources in Chinese, those in English are more mature and abundant. To solve this problem, Wenpeng Lu. et al. [117] proposes a graph-based Chinese WSD method with multi-knowledge integration. Their promising results were

obtained with comprehensive SemEval dataset experimentation on a graph model that combines multiple Chinese and English knowledge resources through word meaning mapping.

Similar to English WSD, Chinese WSD benefits from the use of word embeddings such as Word2Vec or GloVe. These embeddings capture semantic relationships between words and can be used to enhance the representation of Chinese words in a continuous vector space. Due to poor coverage of Chinese by BabelNet, HowNet [118] has been used in a parallel line of study as both a sense inventory and lexical knowledge base for Chinese WSD. HowNet is a widely used Chinese lexical knowledge base. Zhang et al. [119] introduced a novel approach in 2022, integrating monolingual contextual data from a neural language model, bilingual information from machine translation, and sense translation data from HowNet. This approach is a departure from traditional HowNet-based WSD methods.Yiming Cui et al. [120] proposed Pre-Training With Whole Word Masking for Chinese BERT, and their model, MacBERT, has shown state-of-the-art performance in various NLP tasks.

Recent supervised neural WSD algorithms improve performance by making use of a lexical knowledge base, such as by integrating definitions [121]. In 2019, L. Huang et al. build context-gloss pairs and suggest three BERT-based models for WSD. They fine-tune the pre-trained BERT model to produce new, state-of-the-art outcomes on the WSD problem [122]. The study claims that graph-based models are effective in Chinese WSD because they generate graphs with nodes representing words or senses and edges representing semantic links.

2.4.3 WSD in Arabic Language

Since every language has its idioms, grammatical structures, and word usage patterns, it is difficult to directly apply WSD techniques across languages. Word sense disambiguation is a global problem that crosses linguistic boundaries, with

Arabic and other Semitic languages facing especially difficult problems.

The lack of diacritics in many digital texts is a major cause of word ambiguity in Arabic, allowing the same word to appear in many senses [111]. It allows identical words to take on various meanings; without diacritics, they will be the same word. Furthermore, some Arabic prefixes are closely connected to Arabic words and can be regarded as words in the majority of Latin languages, which is to blame for the agglutinative nature of the Arabic language [123].

Arabic words have contextual variations according to diacritical markings, part-of-speech (POS) characteristics, and the context in which they occur (Habash, 2007 [124]). Furthermore, Zitouni in 2014 [125] emphasised that the lack of vowel markers in Arabic script adds considerably to the language's intrinsic morphological ambiguity. Remarkably, a 2002 study [126] by Debili and colleagues found that Arabic texts that have vocalization marks (i.e., marks that indicate vowel sounds) typically had less ambiguous words (43%) than texts that do not have vocalization markers (72%).Using thematic words from a specific context, Sakhr researcher Achraf Chalabi created a new word sense disambiguation algorithm in 1998 that has been implemented in the system's Arabic-to-English computer-aided translation [127].

Arabic language still suffers from the lack of linguistic resources [123] such as lexicons, thesauri, tagged corpora [128], and standardized test collections [129]. furthermore, due to this language's extensive use of inflection, complex morphology in Arabic has a significant impact on IR systems [130]. The richness of Arabic texts and the **free word order** phenomenon are the main issues with statistical methods for Arabic word sense disambiguation. The most prominent supervised approach for Arabic WSD is the work by Habash et al. [131]. Their achievement of statistically meaningful outcomes on well-known benchmark datasets, such the Arabic Treebank, sets them apart from the others.

A recent study of Arabic word sense disambiguation highlights some key un-

resolved challenges that need more research. Two important areas are domain-specific Arabic word sense disambiguation and multi- and cross-lingual disambiguation for Arabic. These topics showcase both current difficulties and chances to advance Arabic word sense disambiguation through further examination.

2.4.4 WSD in Indian languages

India, which is known for its linguistic diversity, is home to 122 major languages and 1599 other languages, according to M. Priya et al. [132]. Approximately 70% of people speak Indo-Aryan languages, and 19% speak Dravidian languages, which include Bengali, Marathi, Telugu, Tamil, Gujarati, Kannada, and Malayalam. These languages are distinguished by their morphological richness and agglutinative structure. Word Sense Disambiguation has been used more frequently in English and other European languages than in Indian languages, according to Alon et al., [133]. The large variety of morphological inflections present in Indian languages, together with the lack of machine-readable dictionaries, word sense inventories, and other knowledge resources required for WSD algorithms in these languages, are the reasons for the suboptimal efficacy of WSD methods.

One of the morphologically rich languages in India is the **Marathi** language [134]. As per the research conducted by Ujwalla Gawande et al., [134], a novel approach to WSD for the Marathi language has been proposed using machine learning techniques. The other language is **Manipuri**, which is distinct from other Indian languages in terms of both syntactic and semantic features and is spoken in a particular geographical area [133].In a pioneering work by Richard Singh and K. Ghosh [135], a novel architecture was proposed in 2013 specifically for the Manipuri Language.

Malayalam is a Dravidian language, primarily spoken in the southern Indian state of Kerala. Rosna P. Haroon proposed Malayalam WSD in 2010 using a knowledge-based approach and suggested that the approach resulted in poor

accuracy due to the scarcity of corpus languages like Malayalam. In 2016, S. Gopal [136] proposed a supervised Malayalam word sense disambiguation system using the Naive Bayes classifier. Because the quality corpus was used, the accuracy was 90%. Another Indian language with a rich morphological system is **Punjabi** [133]. In 2020, VP Singh et al. [53] recommended using deep learning approaches for word sense disambiguation in the Punjabi language. Their investigation demonstrates the effectiveness of these methods for correctly distinguishing between word senses in the context of Punjabi.

Challenges in WSD for low-resource Indian languages include complex morphology and limited resources. Positive progress is seen with deep learning models pre-trained on large datasets, offering effective solutions. Despite linguistic diversity and resource constraints, innovative techniques are driving notable advancements, especially in languages like Marathi, Manipuri, Malayalam, and Punjabi.

2.4.5 WSD in Amharic Language

Amharic is the official language of Ethiopia which, is the second most spoken semantic language group. Due to the morphological richness of the Amharic language, the variety of suffixes, prefixes, and infixes, makes it difficult for word sense disambiguation tasks.

Generally, word sense ambiguity in Amharic is classified as Lexical ambiguity, phonological Ambiguity, structural ambiguity, syntactical ambiguity, and referential ambiguity as discussed below [137, 138, 139, 39]. **Lexical Ambiguity**, occurs when a word has multiple meanings. **Phonological Ambiguity**: Amharic word sense disambiguation faces a significant obstacle known as phonological ambiguity. This phenomenon occurs when multiple words in Amharic share identical pronunciations but carry distinct meanings. The complexity of the Amharic sound system contributes to this issue, as it encompasses a wide range of consonants and vowels that possess nuanced differences in pronunciation.

Semantic ambiguity [40], on the other hand, falls into another category. It arises when words possess multiple meanings that are either related or unrelated to each other. This type of ambiguity is commonly encountered in idiomatic expressions, metaphorical language, and polysemic constructs.

Another class of ambiguity in Amharic is **structural ambiguity** [138]. This type of ambiguity arises from the ability to change the word order and have multiple possible positions or arrangements within the grammatical structure of a sentence. Syntactic ambiguity allows for conveying more than one meaning due to these variations in sentence structure. The ambiguity class is **Referential ambiguity** which is a well-known phenomenon in natural language processing and refers to cases where a word or phrase has multiple possible referents [140].

Structural and syntactical Ambiguity remains unsolved in the case of the Amharic language [138, 39]. The researcher has been exploring and attempting WSD which relies on knowledge-based approaches which are labeled by their own, so the model was unable to learn by itself.

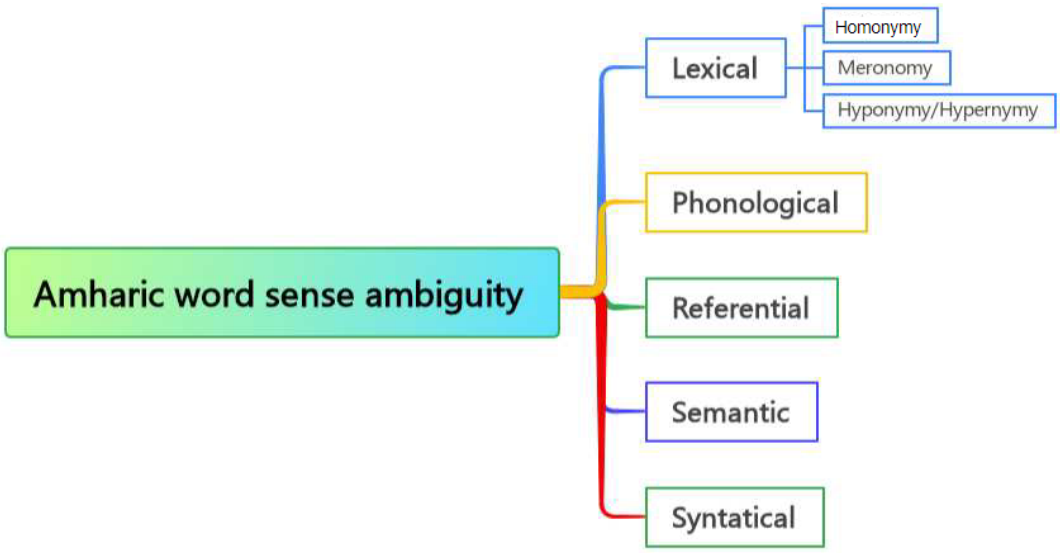


Figure 2.10: Word sense Ambiguity type summary

Amharic Word Sense Disambiguation research has Initial attempts relied on semantic vector analysis, with Kassie 2009 [140] achieving promising results. Supervised machine learning approaches then took center stage, with Solomon 2010

[137] reaching a higher accuracy.

Since then, advancements have included incorporating ensemble classifiers by Hagere [141] and exploring semi-supervised learning Wassie in 2014 [139], demonstrating the continuous development in Amharic WSD. In 2019 [138], Mulugeta developed a system for Amharic WSD using the Amharic WordNet. Unlike previous studies that mainly focused on verbs, Mulugeta's system aimed to cover a broader range of word types, including verbs, nouns, adverbs, and adjectives.

In 2021, Senay [40] and colleagues presented an Amharic Word Sense Disambiguation system based on deep learning. They trained three different deep learning models (LSTM, CNN, and Bi-LSTM) on a dataset containing 159 ambiguous words, 1214 synsets, and 2164 sentences. The models achieved accuracies of 94%, 95%, and 96% for LSTM, CNN, and Bi-LSTM, respectively. In 2022 [39], NEIMA conducted an Amharic WSD study using Amharic RoBERTa (AmRoBERTa), a transformer-based contextual embedding technique. This study demonstrated the potential for precise WSD in Amharic through advanced language models and transfer learning.

Table 2.10: Summary of WSD Papers in Different Languages

Paper	Year	Technique	Dataset	Language
Zhi-Zhuo & He-Yan [116]	2012	Graph	Own	Chinese
XR Sun et al. [115]	2017	LSTM	Own	Chinese
Wenpeng Lu. et al. [117]	2019	Graph	BabelNet	Chinese
L. Huang et al. [122]	2019	BERT	Own	Chinese
Yiming Cui et al. [120]	2019	BERT	Own	Chinese
C. Hadiwinoto et al. [36]	2019	Pre-trained	OntoNotes	Chinese
B Hou et al. [118]	2020	Unsupervised	HowNet	Chinese
Y. Cui et al. [120]	2021	Pretrained	Own	Chinese
X. Zhang et al. [119]	2022	Novel	HowNet	Chinese
I. Bounhas et al. [129]	2011	Knowledge	ArabOnto	Arabic
Debili et al. [126]	2002	Knowledge	Corpus	Arabic
Habash et al. [124]	2007	Knowledge	Corpus	Arabic
Habash et al. [131]	2013	Knowledge	Own	Arabic
S. Elmougy et al. [111]	2008	Naive Bayes	Own	Arabic
Kharate & Patil [142]	2021	Knowledge	WordNet	Marathi
U. Gawande et al. [134]	2023	Novel	WordNet	Marathi
R. Singh & K. Ghosh [135]	2013	Decision Tree	Own	Manipuri
S. Gopal [136]	2016	Naive Bayes	Corpora	Malayalam
J. Sarmah & S.K. Sarma [77]	2016	Decision tree	Own	Assamese
VP Singh et al. [53]	2020	Deep learning	Corpus	Punjabi
Kassie [140]	2009	Knowledge	Own	Amharic
Solomon [137]	2010	Supervised	Own	Amharic
Assemu [143]	2011	Unsupervised	Own	Amharic
Hagerie [141]	2013	Ensemble	Own	Amharic
Wassie [139]	2014	Semi-supervised	Own	Amharic
Mulugeta [138]	2019	Deep learning	Own	Amharic
Senay et al. [40]	2021	Deep learning	Own	Amharic
NEIMA [39]	2022	Deep learning	Own	Amharic

2.4.6 Challenges in WSD

The complexity and ambiguity inherent in language present numerous challenges for WSD. **Complex sentences** containing multiple ambiguous words are particularly ambiguous, requiring additional contextual information for accurate disambiguation. **Contextual dependency** further complicates WSD, as the meaning of words is heavily influenced by the surrounding context, often necessitating the analysis of entire sentences. **Fine-grained** sense differentiation is another major challenge in WSD, which can significantly impact text interpretation. Additionally, the presence of **polysemy** and **homophones** poses challenges, as these involve words with multiple meanings or different senses but identical forms.

Table 2.11: Summary of WSD Challenges across Languages

Language	**Challenges**	**Unique Characteristics**	**Open Issues**
English	Ambiguity in Polysemous Words	Morphologically poor, Extensive use of idioms	sense granularity, fine-grained senses
Chinese	Character level ambiguity, Lack of Clear Word Borders	Unique character-based writing system(logographic in nature)	Cross-lingual sense alignment, Improving word segmentation
Indian	Lack of Annotated Data, Dialectal Variations	Rich linguistic diversity, Script variations	Development of dialect-specific models, Semantic complexity
Arabic	contextual variations according to diacritical markings	Rich morphological structure	Building standardized evaluation, context-aware models
Italiano	Lexical ambiguity	Rich morphological structure	context-aware models, lack of dataset
Amharic	Lack of standard dataset and Limited Parallel Corpora	Morphologically rich and the language characterized by a lot of dialect	Development of cross-lingual model

2.5 Related Works

We provide relevant research that supports our thesis in this section. In particular, research focuses on contextual embedding strategies, various attention mechanisms, and how to incorporate these approaches into BiLSTM models for WSD. We have introduced a neural network seq2seq model and a supervised technique, and we have conducted comprehensive comparisons with the progress and drawbacks of word sense disambiguation.

According to related research, by demonstrating the ability to handle long-range dependency, the **seq2seq** and **BiLSTM** models reach state-of-the-art performance in word sense disambiguation. In addition, new attention mechanisms have been used, and word embedding (bert-embedding) from conventional to large language models has greatly enhanced model performance.

Then We compare baseline paper architecture with our proposed model and clearly define gaps both in English and Amharic. Finally, we present a comparison of our proposed model against baseline papers with some desirable properties

2.5.1 WSD Using BiLSTM Models

Despite the longstanding challenges in the field of Word Sense Disambiguation, the introduction of neural networks has sparked significant transformation and gained much attention in recent years[144]. Specifically, RNNs are good at identifying long-term dependencies within sequences[145, 144], which is the most important thing in word sense disambiguation. BiLSTM model is uniquely suited to handle the complex and extensive dependencies that define and represent the use of word senses [95, 144, 146].

Due to the persistent challenge of the exploding or vanishing gradient problem in RNNs, LSTM [97] and GRU [98] were introduced as robust solutions. Then BiLSTM, which is an extension of the LSTM network, was introduced by Graves et

al. in 2005 [147]. By employing bidirectional processing, the Bi-LSTM captures more comprehensive information from sentences, surpassing the capabilities of the standard LSTM network [32]. This makes the Bi-LSTM well-suited for tasks that involve bidirectional context dependence.

In addition to BiLSTM, we explored various baseline methods to build a strong foundation for our WSD research. The initial work in word sense disambiguation often regarded as the baseline approach is the **Most Frequent Sense** approach, which selects the sense that occurs most frequently in the training data. However, this method has limitations in its generalizability and its capability to handle unseen data. Basile et al. [148] introduced Leskext+emb, a method inspired by the classic Lesk algorithm (Lesk, [13]). It uses a word similarity function in a distributional semantic space to match glosses with their contextual usage. **UKB** by Aggrie et al. [149]: Leverages the UKB knowledge base for context-aware disambiguation, overcoming MFS limitations. **Babelfy** by Moro et al. [67]: Utilizes BabelNet's semantic network for both WSD and Entity Linking, similar to UKB.

Supervised Approach: The Supervised method represents an advancement over knowledge-based approaches. For example, **Context2Vec** by Melamud et al., 2016[80] and **IMS** by Zhi and Ng (2010) [81] utilize a linear SVM classifier and features like POS tags, nearby words, and local collocations within a limited window around the target word. Another approach, **IMS+emb**, developed by Iacobacci et al. [82], enhances IMS by incorporating word embeddings as features, which has proven highly effective across various WSD datasets.

Neural Netwerk: **Seq2Seq** model by M. Ahmed in 2018 [43] and its extensions like **Seq2Seq + att.** have shown promise. Similarly, Kaageback and Salomonsson developed a **BiLSTM** model for WSD in 2016 [145]. These models, along with **Bi-LSTM+att.+LEX** and **Bi-LSTM+att.+LEX+POS** by Raganato et al. 2017a[42], transforming WSD into a sequence learning problem

and incorporating multi-task learning setups.

Using Attention in WSD: The transformative impact of attention mechanisms in NLP has led to their widespread adoption in word sense disambiguation. Self-attention [37, 32, 146] and multi-head attention [150, 151, 146] have shown promising results in WSD tasks.

Vaswani et al. [23] introduced multi-head attention, applied in various NLP tasks, including WSD. Domhan [152] emphasized the effectiveness of multilayer attention in NMT, showing performance gains by using attention across multiple layers. Multi-head attention combines linear projected attention and concatenates the results for the decoder network, contrasting with single-head attention.

In 2018, Ahmad et al. [153] employed multiple attention mechanisms for POS, bigram, and words, combining these weights for significant improvement. However, the success of attention mechanisms relies on their application to the right purpose and layer. Tang et al. [154] discovered in 2018 that attention mechanisms tend to focus more on the ambiguous noun itself than on context tokens, unlike other nouns.

BERT Embedding: Research has explored traditional word embedding approaches like Word2vec and GloVe. Kang et al. [155] extended the Word2vec model to capture fine-grained meaning in Korean. Orkphol and Yang [156] improved syntactical and semantic features using Word2vec. However, these traditional embeddings have limitations in capturing contextual variations in meaning. This limitation has led to the development of deep contextualized models like ELMO and BERT, trained on extensive corpora to address this challenge.

ELMO [20] and BERT [14] improve upon traditional word embeddings by considering contextual information. Kutuzov and Kuzmenko [157] leveraged ELMO embeddings for word sense disambiguation in English and Russian, showing the importance of lemmatization training for morphologically rich languages like Russian. Additionally, many studies have used BERT for word sense disambiguation,

either through fine-tuning for specific tasks [34, 24] or incorporating sense embeddings [158, 159]. While these approaches have achieved state-of-the-art results, further refinement is needed.

2.5.2 Baseline

Sequence-to-sequence model specially BiLSTM [145, 43, 42, 146], now gained popularity in word sense disambiguation and machine translation [98, 160]tasks.

We adopt the paper suggested by Raganato et al. [42] as a baseline, providing a solid framework for word sense disambiguation. They employ encoder-decoder architecture with BiLSTM layer in both modules. The **encoder** utilizes an embedding layer to convert input sequences into vector representations. These vectors are then processed through one or more bidirectional LSTMs, resulting in a context vector that captures the semantic meaning of the entire input sentence. The **decoder**, also employing bidirectional LSTMs, leverages the context vector and its hidden state to generate the output sequence word by word. Unlike some models, the decoder here has access to the context vector at every step, allowing for consideration of the entire input during translation. Finally, they apply a softmax layer to convert the decoder's output into probabilities for each word in the vocabulary.

Our model takes inspiration from the work of Raganato et al.[42], but we have significantly enhanced the architecture to achieve superior performance in Word Sense Disambiguation. The initial step of our model involves processing the input sentence to create a comprehensive vector representation. Utilizing word embedding allows us to encode the meaning of each word independently, while contextual embedding, employing BERT, enhances our understanding of the interplay between these words in the context of the sentence.

The vectorized sentence is processed by a BiLSTM layer, comprising two directional LSTMs, enabling the model to capture forward and backward depen-

dencies for a comprehensive context. To enhance this, local attention (word-level) is applied to the first BiLSTM layer's output, focusing the model's attention on specific words for key information extraction.

Then we apply another attention-based stacked BiLSTM layer, which is inspired by the work of Sun et al. [32]. Stacked Bidirectional LSTM in a sense, contains multiple layers of BiLSTM cells stacked on top of each other. Each layer processes the output of the previous layer, capturing increasingly abstract representations of the input sequence. Zobaed et al. [161] also propose a sense-pick method by using a stacked BiLSTM approach and achieved a 3.5 F1-score over baseline models.

The output of the local attention mechanism is passed through a second BiLSTM layer to refine the contextual representation. Then, global attention is applied over the entire sentence to capture long-range dependencies. The output of the global attention mechanism is integrated with the hidden state of the model through a context and hidden layer. Finally, a softmax layer on a fully connected layer generates a probability distribution for each sense, enabling the model to predict the most likely sense for the target word based on accumulated contextual information.

By incorporating these elements, our model builds upon the strengths of the baseline paper while introducing several key enhancements:

- Enhanced Contextual Understanding: The combination of local and global attention mechanisms enables the model to capture both fine-grained and broader contextual information, leading to a more comprehensive understanding of the sentence.

- Improved Long-Range Dependency Handling: The use of BiLSTM layers and global attention allows the model to effectively handle long-range dependencies, which are crucial for accurate disambiguation in complex sen-

tences.

- Adapted our model to Amharic and Italian: Our model demonstrates the effectiveness of the proposed model to improve WSD performance in Amharic and Italian languages.

Table 2.12: Comparison of Baseline Model and Proposed Model

Feature	**Baseline Model Raganato et al.**[42]	**Proposed Model**
Architecture	Encoder-decoder with BiLSTM layers	Encoder only with stacked BiLSTM layers, local attention, global attention, and context and hidden layer
Encoder	BiLSTM with embedding layer	Stacked BiLSTM with BERT embedding layer, local attention, and global attention
Decoder	BiLSTM with access to context vector at every step	Not applicable (encoder-only model)
Output	Softmax layer for word probabilities	Softmax layer for word probabilities
Strengths	Solid framework, captures semantic meaning of input sentence, considers entire input during translation	Improved contextual understanding through local and global attention, enhanced long-range dependency handling, and demonstrates cross-lingual WSD
Limitations	Unable to capture contextual information, limited ability to handle complex linguistic contexts	It depends on labeled data

2.5.3 Identifing Reseach Gaps

Existing models mostly focus on either syntactic features (sentence structure) [38] or semantic [6] aspects (word meaning) independently. They don't effectively capture the complex interplay between these two aspects, which is crucial for accurate disambiguation. While attention mechanisms are used to focus on

ambiguous words, they primarily target the word itself [154], not the broader context and relationships between words in the sentence. This restricts the model's ability to fully understand the surrounding context.

Table 2.13: Research Gaps and Desirable Properties

Research Gap	**Desirable Property**
▶ Syntactical and semantical ambiguity has been explored, but it remains challenging. Models need to be more contextual to capture linguistic information effectively [6, 42, 38].	▶ Develop models that are more contextual to capture linguistic information and address the challenges of ambiguity.
▶ previous model utilizes Attention mechanisms primarily emphasizing ambiguous words, neglecting contextual relations among words [162].	▶ Develop attention mechanisms that consider contextual relations among words for better disambiguation.
▶ Models struggle with complex sentence structures and words with multiple meanings [87, 68].	▶ Enhance models to handle complex sentence structures and multiple meanings more effectively.

Note: Each row presents a specific challenge indicated by ▶ symbol shows the challenge faced in current WSD methodologies, such as the need for more contextual models to capture linguistic information effectively. The corresponding desirable property indicated by ▶ outlines the goal of developing models that address these challenges, aiming for improved disambiguation accuracy.

The table 2.13 below highlights key research gaps and desirable properties in English WSD, offering a concise overview of areas needing advancement. It serves as a guide for future research to develop more effective WSD solutions.

For **Amharic language** word sense disambiguation has covered the category of hypernym and hyponym [40, 138, 39]as shown in Table 2.14. However, there are still uncovered areas, such as syntactical ambiguity, semantical ambiguity, and phonological ambiguity. Hypernyms are broader terms that encompass narrower terms, while hyponyms are specific terms categorized under a broader term. Syntactic ambiguity involves multiple interpretations arising from sentence structure, while semantic ambiguity relates to multiple meanings of words or phrases. Phonological ambiguity is a word having the same sound but different meanings.

Table 2.14: Desirable property in Amharic WSD

Category	Status
Hypernyms	✓
Hyponyms	✓
Syntactical Ambiguity	✗
Semantical Ambiguity	✗
Phonological Ambiguity	✗

Note:✓ indicate research done and symbol ✗uncovered area

Table 2.15: Comparisons different paper with our approach

	Features			
Approach	Attention	syntactical	semantic	long-range dependency
MFS	✗	✓	✗	✗
Leskext+emb [148]	✗	✓	✗	✗
UKB [149]	✗	✓	✗	✗
Babelfy [67]	✗	✓	✗	✗
Context2Vec [80]	✗	✗	✓	✗
IMS+emb [82]	✗	✗	✓	✗
Seq2Seq [43]	✓	✗	✓	✗
BiLSTM att+LEX [42]	✓	✗	✓	✗
BERT	✓	✓	✓	✗
Ours	✓	✓	✓	✓

Note: For each feature, a green checkmark (✓) indicates its inclusion, while a red (✗) indicates its absence. Our approach stands out by incorporating all these features, demonstrating its comprehensive approach to word sense disambiguation compared to existing methods.

Table 2.15 compares different approaches, including our own, based on key features crucial for WSD. Our approach, along with Seq2Seq and BiLSTM att+LEX, incorporates attention mechanisms, which enable the model to focus on specific parts of the input sequence, enhancing its ability to disambiguate word senses in context. However, most approaches, such as MFS, Leskext+emb, UKB, and Babelfy, do not explicitly include syntactical analysis, potentially limiting their capacity to capture the structural aspects of language essential for WSD. On the other hand, our approach, along with Context2Vec and IMS+emb, includes semantic analysis, allowing for a deeper understanding of the meaning of words and their relationships, which can significantly improve disambiguation accuracy.

Moreover, our approach stands out for explicitly addressing long-range dependencies, a feature crucial for capturing relationships between words that are distant from each other in the sentence. Overall, our approach encompasses all these features, making it a comprehensive and promising solution for WSD compared to existing methods.

2.6 Summary

In this section, the reviewed literature emphasizes the important role played by BiL-STM models in Word Sense Disambiguation (WSD). Specifically, when BiL-STM models are combined with attention mechanisms and contextual models, they have shown to be highly effective. Various studies have showcased the ability of BiLSTM models to capture contextual information and long-range dependencies, which are essential for accurately disambiguating word senses.

Raganato [42] a unified evaluation framework Method: The Ragnato unified evaluation framework method is a technique that combines multiple sources of information, such as lexical resources, semantic networks, and context, to disambiguate word senses. This method has been shown to improve WSD performance by leveraging diverse sources of information to make more informed sense distinctions.

GlossBERT: GlossBERT [34] is a variant of the BERT model that incorporates gloss information from lexical resources, such as WordNet, into the pretraining process. By augmenting the contextual embeddings with gloss information, GlossBERT aims to improve the representation of word senses and enhance WSD performance.

Attention Mechanisms: Attention mechanisms have been applied to WSD to improve the model's ability to focus on relevant parts of the input sequence. By attending to informative context words, attention mechanisms help disambiguate word senses more effectively, leading to improved WSD performance. Despite

that, applying hierarchical attention shows amazing performance.

BiLSTM: BiLSTM models show significant performance in the field of word sense disambiguation, due to their ability to capture increasingly complex patterns in the input sequence. By stacking multiple layers [32], these models can learn hierarchical representations of the input, which has been shown to improve WSD performance.

Contextual Embeddings like BERT: Contextual embeddings, such as those generated by BERT, capture the meaning of words based on their surrounding context. By leveraging contextual embeddings, WSD models can better disambiguate word senses by considering the broader context in which words appear.

These advancements collectively illustrate the continuous endeavors to enhance the performance of Word Sense Disambiguation by integrating various methods and techniques. Researchers are consistently combining approaches such as attention mechanisms, BiLSTM, and contextual embeddings like BERT.

The primary objective of the proposed approach is to bridge the existing gap in addressing semantical ambiguity in Word Sense Disambiguation by improving attention mechanisms. Previous studies have incorporated attention mechanisms, but their effectiveness has been constrained by their limited capability to consider the broader sense context of words. Instead, these mechanisms have predominantly focused on the individual targeted word. This restriction impairs the model's accuracy in disambiguating word senses, particularly when faced with instances of syntactical and semantical ambiguity.

The proposed approach seeks to enhance attention mechanisms in WSD by integrating BERT embeddings. BERT embeddings are contextual embeddings that capture the semantic meaning of words by considering their surrounding context. By incorporating BERT embeddings, the proposed approach aims to enhance the model's contextual understanding, enabling it to more effectively disambiguate word senses within syntactically and semantically ambiguous contexts.

The studies reviewed collectively justify the proposed approach of enhancing Word Sense Disambiguation through the utilization of Hierarchical Attention and Semantic Integration Using the BiLSTM Model. By capitalizing on the advantages offered by BiLSTM models, attention mechanisms, and contextual embeddings, the proposed approach strives to enhance the accuracy and resilience of WSD systems. Ultimately, this contributes to the progress of the broader field of natural language processing.

Chapter 3

Methodology

3.1 Introduction

In this section, a methodology is introduced for the creation and evaluation of hierarchical attention mechanisms in word sense disambiguation. The methodology incorporates contextual embeddings on BiLSTM and is divided into four sub-sections: data collection, preprocessing and feature extraction, model architecture, and experimentation and evaluation. This comprehensive methodology offers a clear and structured guide for conducting research in the field of WSD.

3.2 Research Methodology

We use the Design Science Research (DSR) process [163] as the main research approach in this work. DSR is especially well-suited to this research since it concentrates on developing and assessing artifacts to address particular problems. The DSR process consists of several iterative steps, including problem identification and motivation, definition of objectives for a solution, design and development of the artifact, demonstration of its utility, and evaluation of the artifact's effectiveness. The iterative nature of DSR allows for refinement and improvement of the artifact based on feedback and evaluation results. So throughout this thesis, we follow the following steps, as presented in the figure.

- **Problem Identification and Motivation**: As we first conduct a preliminary literature review, to help us clearly define our problem statement as

it does in 1.2. On follow-up, we formulate three research questions in 1.3, which help us to frame our specific objective.

- **Objective Definition**: Building on the identified problem, the research aims to enhance hierarchical attention and semantic integration with BiLSTM model, as stated in 1.4.1. We subsequently established specific objectives 1.4.2 that subdivided the overarching goal into more manageable parts, providing clear guidance for the specific steps we needed to undertake.

- **Design and Development**: The core of the research involves the design and development of the proposed enhancing WSD using hierarchical attention-based and semantic integration with BiLSTM model. The model architecture consists of multiple components, including word embeddings, a hierarchical attention mechanism, a BiLSTM layer, and a softmax classifier for sense prediction. The design of each component is informed by existing literature and best practices in WSD.

- **Evaluation**: The model was evaluated using a benchmark dataset, with performance metrics like accuracy, precision, recall, and F1 score. It outperformed baseline models and achieved competitive performance compared to state-of-the-art methods.

- **Knowlege-contribution**: By integrating the hierarchical attention mechanism into a BiLSTM architecture, this research demonstrates the effectiveness of combining attention mechanisms with deep learning models for WSD. We stated our contribution in 1.6.

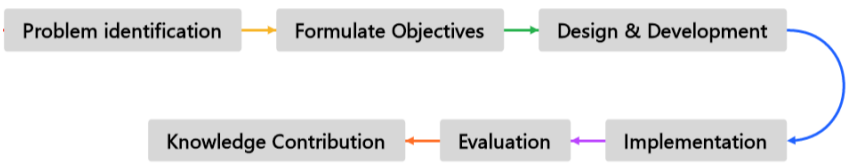


Figure 3.1: DSR process for WSD enhancement in this thesis

3.3 Data collection

This section provides a comprehensive overview of the data collection process for the SemCor 3.0 dataset, which serves as the primary dataset for training and evaluating our Word Sense Disambiguation research.

SemCor 3.0, a corpus created by Princeton University, is an invaluable resource for conducting research in Word Sense Disambiguation. Derived from the Brown Corpus, it comprises sentences wherein every word has been painstakingly annotated with its respective sense from WordNet. The collection of the SemCor 3.0 dataset was carried out during the Senseval-3 lexical sample task. Annotators were provided with sentences sourced from the Brown Corpus and tasked with annotating the sense of each content word (nouns, verbs, adjectives, and adverbs) based on WordNet.

The research community highly values the SemCor 3.0 dataset for evaluating WSD systems, thanks to its extensive annotations and diverse text sources. By incorporating this dataset, we establish the credibility and reliability of our approach, which employs hierarchical attention and contextual embedding to enhance WSD. Leveraging SemCor 3.0 enhances the validity and dependability of our proposed methodology, making a valuable contribution to the advancement of WSD techniques.

Table 3.1: SemCor Dataset Statistics

Attribute	**Value**
# Corpus	SemCor (Miller et al., 1994)[105]
# Documents	352
# Annotations	226040
Annotation Type	WordNet Senses
Format	eXtensible Markup Language (XML)

Originally, the SemCor dataset was an XML file, containing sentences and instances with target words. The other is the Gold Key text file, which contains ground truth sense keys for the target words, serving as a reference for evaluating

model performance. So in favor of our training, we extract sentences with their target words. Important information is also mapped from XML files, like the pos and lemma of target words.

The data collection process involves several key steps:

1. **Loading and Parsing Data:** Parse the SemCor XML file to extract sentences, target words, and their associated information. Additionally, load the ground truth sense keys from the separate text file.

2. **WordNet Integration:**

   (a) **Sense Keys:** For each target word, acquire potential sense keys representing different meanings from WordNet.

   (b) **Glosses:** Obtain definitions (glosses) for each sense key from WordNet, providing context for the potential meanings.

   (c) **Morphological Handling:** Account for morphological variations of the target word to ensure accurate information retrieval from WordNet.

3. **Data Combination and Augmentation:**

   (a) Map the correct sense keys from the gold data.

   (b) Randomly choose and add additional sense keys up to a predefined maximum to enrich the data

   (c) Shuffle the order of sense keys and their corresponding glosses

4. **Sentence Augmentation:** Mark the target words within the original sentence with "[TGT]" to highlight their position.

5. **Data Storage:** Write the processed data, to a CSV file, creating a structured dataset for WSD tasks.

Algorithm 2 Extracting Information from XML for WSD

```
procedure ExtractInfo(XML file, WordNet)
    for all sentence in root.iter('sentence') do
        for all instance in sentence.iter('instance') do
            instance_id ← instance.attrib.get('id','')
            lemma ← instance.attrib.get('lemma','')
            pos ← instance.attrib.get('pos','')
            sense_id ← instance.attrib.get('sense','')
            context ← ' '.join([wf.text.strip() for wf in sentence.iter('wf')])
            synsets ← wn.synsets(lemma)
            gloss ← ""
            for all synset in synsets do
                if synset.name() == sense_id then
                    gloss ← synset.definition()
                    break
                end if
            end for
            writer.writerow({'ID' : instance_id,'Word' : lemma,'Sense' : sense_id,
    'POS' : pos,'Sentence' : context,'Gloss' : gloss})
        end for
    end for
end procedure
```

The collected data for word sense disambiguation is organized and stored in a comma-separated values (CSV) file with specific columns:

- **id**: A unique identifier for each sentence in the dataset.
- **sentence**: The original sentence with the target words explicitly marked as "[TGT]" for clarity.
- **sense_keys**: A list containing various potential sense keys, representing different meanings the target word could have in the context of the sentence.
- **glosses**: Definitions (glosses) retrieved from WordNet for each sense key, providing additional information and context about the potential meanings.
- **target_words**: Indices corresponding to the correct sense keys for the target words within the sentence. This information serves as the ground truth for evaluating the performance of a WSD model.

Through the data collection process, a dataset for WSD is constructed by utilizing SemCor data, ground truth sense keys, and WordNet information. By amalgamating sense keys, glosses, and context extracted from the original sentences, the resulting dataset establishes a robust foundation for training and evaluating our WSD model. This comprehensive dataset empowers us to develop and assess the performance of our WSD model effectively.

3.3.1 Amharic dataset

Because there was a scarcity of existing benchmark datasets for Amharic Word Sense Disambiguation, we took the initiative to create our own dataset using diverse sources such as Reporter-News, BBC Amharic, EPA (Ethiopian Press Agency), Soccer Ethiopia, and the Amharic Bible. To ensure diversity in the dataset, we incorporated various domains such as health, sports, business, and politics.

Data Collection

During the data collection phase, approximately 50,000 sentences were gathered from the mentioned sources. This extensive collection ensures that the dataset captures a wide range of language use across different contexts and topics. The dataset comprises annotated sentences containing target words and their associated senses, offering a comprehensive resource for training and assessing our WSD model.

```
1
2 def fetch_target_words(target_words):
3
4     return target_words
5
6 def split_into_sentences(text):
7
```

```
    amharic_punctuation = r"::|:|?|!"

    # Segmenting the text based on punctuation marks
    sentences = re.split(amharic_punctuation, text)

    # Filtering out empty strings and trimming leading/trailing whitespace
    sentences = [sentence.strip() for sentence in sentences if sentence.strip()]

    return sentences

if __name__ == '__main__':
    target_words = ['word1', 'word2', 'word3']  # Add more target words

    corpus_directory = 'amharic_corpus/amharic_data'  # Specify the directory containing the corpus
    output_csv_file = 'output.csv'  # Specify the desired output file name

    target_words = fetch_target_words(target_words)

    with open(output_csv_file, 'w', newline='', encoding='utf-8') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(['ambiguous_word', 'sentence'])  # Writing the header row

        processed_sentences = set()  # Storing processed sentences for uniqueness

        for filename in os.listdir(corpus_directory):
```

```
            if filename.endswith('.txt'):
                corpus_file = os.path.join(corpus_directory, filename)
                with open(corpus_file, 'r', encoding='utf-8') as f:
                    for paragraph in f.read().strip().split('\n\n'):
                        for sentence in split_into_sentences(paragraph):
                            for target_word in target_words:
                                # Checking if the target word appears as a standalone word in the sentence
                                if re.search(r'\b' + re.escape(target_word) + r'\b', sentence):
                                    if sentence not in processed_sentences:  # Ensuring uniqueness
                                        writer.writerow([target_word, sentence])
                                        processed_sentences.add(sentence)
                                    break  # Exiting the loop once a target word is found

 print(f'Extracted unique sentences with the ambiguous words have been saved to "{output_csv_file}".')
```

Listing 3.1: Data collection sample code

Data Annotation

The annotation process was carried out by our team, with linguists employed for evaluation purposes. The annotated dataset includes sentences containing target words and their associated senses. The steps involved in the annotation process

are:

- **Initial Cleaning**: The raw text data was cleaned to remove non-Amharic characters, HTML tags, and other irrelevant content.
- **Sentence Segmentation**: The cleaned text was segmented into sentences using regular expressions to identify sentence boundaries based on these punctuation marks.
- **Manual Annotation**: Each sentence was manually annotated for word senses. This involved:
  - Identifying ambiguous words within each sentence.
  - Providing sense labels based on context.
  - Adding glosses to clarify each sense.

  In total, 50,000 sentences were manually annotated, covering 220 unique ambiguous words. This comprehensive annotation provides a valuable resource for training and evaluating WSD models. To ensure the quality and accuracy of the annotations, we employed several linguists to evaluate the annotated dataset. Their expertise helped refine the annotations and validate the dataset's reliability.

Data Availability

The dataset is in the process of publication in Science Direct Data in Brief, making it accessible to the wider research community. Once it is published, the dataset will be available for research and educational purposes. Researchers and developers will be able to download the dataset and use it in their projects, with proper attribution to the creators.

3.4 Preprocessing

Text preprocessing plays a crucial role in natural language processing [164], acting as a bridge between raw text and effective machine learning models. In this section, we outline the essential preprocessing steps undertaken to convert unstructured text into a format suitable for word sense disambiguation. In our data preprocessing phase, we focus on key tasks such as vocabulary mapping, tokenization, word embedding, and part-of-speech tagging. For the Amharic dataset, we incorporate a Morphological Analyzer as an additional essential element. Furthermore, we employ AmharicXLMRoberta for tokenization.

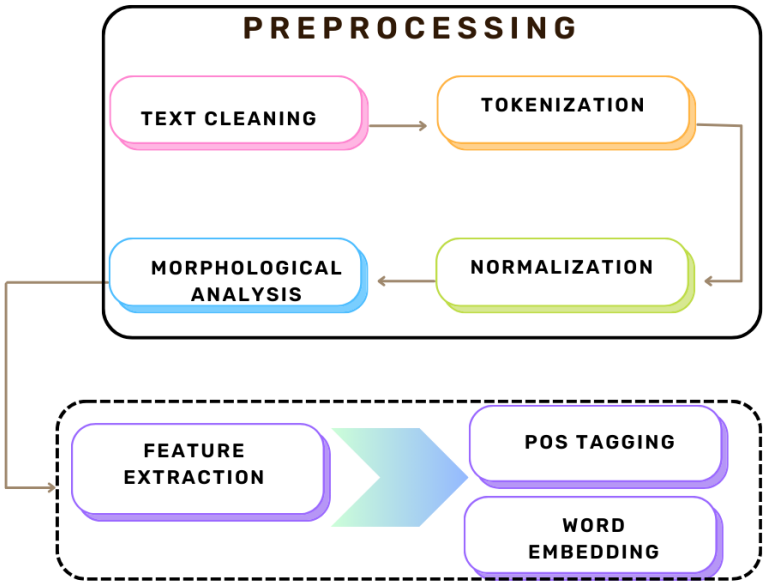


Figure 3.2: Data processing and preparation step

3.4.1 Text Cleaning

- **Lowercasing:** This step ensures consistency in the data by converting all letters to lowercase. This helps machine learning models avoid treating "Hello" and "hello" as different words.

- **Punctuation Removal:** Punctuation marks are removed from the text,

as they often don't carry significant meaning for the model's purpose. This helps the model focus on the core content of the words.

- **Handling Infrequent Words:** Words that appear less frequently than a specified threshold (`min_freq`) are replaced with a special token, typically "<unk>" (unknown). This helps the model avoid overfitting to rare words and improve generalization.

For **Amharic dataset**, we conducted thorough cleaning procedures to ensure data cleanliness. This involved excluding **numbers**, **non-Amharic characters**, **emojis**, and **URLs** from sentences. Given the diverse sources of our data, including various news outlets, this rigorous cleaning was necessary.

3.4.2 Tokenization and Lemmatization

Tokenization and lemmatization are essential preprocessing steps in our data. These steps involve:

- **Tokenization:** Tokenization is the process of segmenting text into individual tokens or words, which is an integral part of our preprocessing pipeline. This step allows for the separation of sentences into discrete units, enabling further analysis and extraction of features. For tokenization, we used **bert-base-uncase** [14] in our English WSD dataset. for our Amharic WSD dataset since **BERT** was not pre-trained in Amharic data thanks to David Adelani we used their model **AmharicXLMRoberta**, which is obtained by fine-tuning **xlm-roberta-base** on Amharic corpus.

- **Lemmatization:** Lemmatization, the process of converting inflected words to their base or canonical form, plays a crucial role in our approach. It assists in consolidating words with similar meanings, enhancing the model's capacity to generalize across various word forms. For example, both **"running"** and **"ran"** would be lemmatized to **"run"**. To ensure consistency

and accuracy in word representation, we utilize the **WordNet lemmatizer** as part of our approach.

In our efforts to enhance the Amharic dataset, we initially explored the utilization of **Hornmorpho**, a morphological analysis tool developed for **Amharic, Tigrigna, and Oromo** languages by M. Gassper [165]. However, we encountered challenges with this tool as it led to changes in meaning and the loss of contextual information. Additionally, the model exhibited shortcomings by providing incomplete words. Consequently, we analyzed these errors and implemented appropriate measures to address them effectively.

3.5 Feature Extraction

In our research, feature engineering plays a crucial role as we aim to extract informative features that enhance our model's performance. To achieve this, we employ various techniques such as utilizing surrounding words within a **context window**, incorporating **part-of-speech tags**, and leveraging **word embeddings**. These techniques enable us to capture valuable information and enrich the model's understanding of the data.

3.5.1 Part-of-Speech Tagging

Part-of-speech tagging is a crucial step in natural language processing, including Word Sense Disambiguation, and assigning grammatical categories (e.g., noun, verb, adjective) to words in a sentence. State-of-the-art algorithms like Hidden Markov Model (HMM), Conditional Random Field (CRF), and neural network-based models such as BiLSTM-CRFs are employed for accurate POS tagging. These algorithms predict precise POS tags by leveraging contextual information and linguistic features.

To efficiently process large text corpora and ensure accurate and efficient POS tagging, we utilize robust libraries like Natural Language Toolkit (NLTK), and Space Polyglot (SpaCy), which offer pre-trained taggers and extensive linguistic resources. For **Amharic dataset** Part-of-Speech (POS) tagging we used General-purpose pre-trained embedding developed by T.D Belay et al. [166], which helps relate amharic words with their different word class information.

3.5.2 Word Embedding

In the field of NLP, word embedding is a critical component, particularly in tasks like WSD. Sense and word embeddings improve how words and their meanings are represented in NLP tasks, including WSD. Our methodology integrates word embedding techniques.

Word Embedding: Word embeddings, like BERT embeddings, encode words into dense vector representations in a continuous space, capturing semantic relationships between words. Utilizing pre-trained models such as BERT allows us to incorporate rich contextual information, enhancing the effectiveness of WSD. For our research, we employed the **Bert-base-uncase** for the English dataset and **AmharicXLMRoberta** for the Amharic dataset.

Sense Embedding: Sense embeddings represent the various senses or meanings associated with a word. These embeddings capture the nuanced semantic distinctions between different senses of a word, facilitating more accurate disambiguation. Techniques for sense embedding may involve clustering algorithms, graph-based methods, or deep learning models trained specifically for sense representation.

3.6 Model Architecture

The primary objective of the proposed model architecture is to tackle the task of Word Sense Disambiguation, which involves predicting the appropriate sense

of a word in a given context. This task holds significant importance in numerous natural language processing applications. By accurately determining word senses, the model enhances text comprehension and boosts performance in downstream tasks like machine translation, information retrieval, and sentiment analysis.

Existing models for WSD have shown promising results but often struggle with capturing the intricate relationships between words in a sentence. Many traditional models rely on handcrafted features or shallow representations, limiting their ability to generalize well across different contexts. Additionally, these models may struggle to handle the high-dimensional nature of language data.

The proposed model architecture addresses these limitations by incorporating deep learning techniques and leveraging pre-trained word embeddings, particularly BERT. By doing so, it can effectively capture contextual information and learn complex representations that capture the nuances of word senses.

3.6.1 Description of Proposed Model Architecture:

The model architecture consists of sequential layers designed to process the input sentence and extract meaningful representations for word sense disambiguation.

Sequential Input

The model receives the input sentence as a sequence of words. The input to the model is a sequential representation of words in a sentence. This input is first passed through a word embedding layer, which utilizes BERT to obtain contextualized word representations. Mathematically, this process can be represented as follows:

Let $s = (w_1, w_2, ..., w_n)$ be a sentence consisting of n words, where w_i represents the i-th word in the sentence. Each word w_i is represented by its BERT embedding $e_i \in \mathbb{R}^d$, where d is the dimensionality of the embedding space.

The sentence s is then represented as a sequence of embeddings $E = (e_1, e_2, ..., e_n)$,

where $E \in \mathbb{R}^{n \times d}$. The model processes this sequence of embeddings E to capture the meaning and contextual information of each word in the sentence.

Word Embedding with BERT

BERT (Bidirectional Encoder Representations from Transformers) embeddings are utilized in the proposed model architecture for Word Sense Disambiguation to enhance the representation of each word in the input sentence. BERT embeddings offer several advantages over traditional word embeddings.

BERT embeddings are contextualized, capturing nuanced meanings and semantic relationships in different contexts. This contrasts with static representations like word2vec or GloVe. we utilize the **bert-base-uncased** model for obtaining these contextualized embeddings. Mathematically, this contextualization can be represented as:

$$e_i = \text{BERT}(w_i, \text{context})$$

BERT embeddings are pre-trained on diverse textual data, enabling comprehensive word representations. This is advantageous over traditional embeddings trained on narrower datasets.

BERT's transformer-based architecture efficiently models long-range dependencies, crucial for capturing complex language patterns in WSD tasks.

The proposed model benefits from BERT's contextualized representations, pre-training on diverse corpora, and ability to model complex patterns, leading to more accurate sense predictions in WSD.

- **BiLSTM Layers:** The Bidirectional Long Short-Term Memory (BiLSTM) layer is employed to process the input sequence of word embeddings from BERT dually, encompassing both the forward and backward directions. This allows the BiLSTM layer to capture both long-term dependencies and

contextual information within the sentence, enhancing the model's understanding of the input data. Mathematically, the BiLSTM layer can be represented as follows:

$$\overrightarrow{h}_t = \text{LSTM}_{\text{forward}}(\text{BERT}(w_t, \text{context})),$$
$$\overleftarrow{h}_t = \text{LSTM}_{\text{backward}}(\text{BERT}(w_t, \text{context})),$$
$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$$

where $\text{LSTM}_{\text{forward}}$ and $\text{LSTM}_{\text{backward}}$ denote the forward and backward LSTM computations, respectively, and h_t denotes concatenation.

Hierarchical Attention

The hierarchical attention mechanism is designed to capture semantic context at different levels of granularity within a sentence. It consists of two levels of attention: word-level attention and sentence-level attention.

Word-level Attention: At the word level, the attention mechanism assigns weights to each word in the sentence based on its relevance to the context. Words more relevant to determining the sense of an ambiguous word receive higher weights. For example, consider the sentence, **He went to the bank to cash out his money**. In this sentence, the word "bank" is ambiguous and could refer to a financial institution or the side of a river. The word **bank** would receive higher attention weights if the context suggests a financial transaction.

Sentence-level Attention: At the sentence level, the attention mechanism aggregates the word-level attention weights to compute a single weight for the entire sentence. This weight represents the overall relevance of the sentence to determining the sense of the ambiguous word. For example, in the sentence **He went to the bank to cash out his money,** the sentence-level attention would consider the entire sentence's context to determine the relevance of each word's

attention weight to the ambiguous word **bank**. So, to perform this hierarchical attention, we employ different attention mechanisms at word level and sentence level, as described below.

- **Local Attention:** A local attention mechanism is utilized to capture the word-to-word relationships among neighboring words. This mechanism enables the model to selectively focus on specific neighboring words that are most pertinent to comprehending the sense of the target word. Mathematically, the local attention mechanism can be described as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

  where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. This mechanism allows the model to attend to a subset of words around the target word, enhancing its ability to capture local context.

- **Global Attention:** To capture the broader context beyond neighboring words and consider the overall contribution of all words in the sentence, a global attention mechanism is employed. For this purpose we use Bahdanau attention a technique that replaces the usual way of processing sentences, allowing the model to focus on different parts of the input sentence as needed. Mathematically, the Bahdanau attention mechanism can be represented as follows:

$$e_{ij} = \mathrm{v}_a^\top \tanh(\mathrm{W}_a[\text{decoder\_hidden}_i, \text{encoder\_hidden}_j])$$
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$
$$\text{context}_i = \sum_j \alpha_{ij} \text{encoder\_hidden}_j$$

where v_a, W_a are learnable parameters, decoder_hidden_i is the hidden state of the decoder at time step i, and encoder_hidden_j is the hidden state of the encoder at time step j. The attention weights α_{ij} determine how much focus should be given to each encoder hidden state when computing the context vector context_i.

- **Hidden and Context Layer:** By combining the outputs from the BiLSTMs and attention mechanisms, this layer generates a more comprehensive representation that incorporates both the local and global context surrounding the target word. This enriched representation captures a holistic understanding of the word by considering its neighboring words as well as the broader context within the entire sentence.

- **Fully Connected Layer:** A fully connected layer maps the combined representation from the previous layer to the output layer. The combined output from the attention mechanisms is passed through a linear layer followed by a softmax activation function to generate the final sense prediction probabilities.

$$\text{out} = \text{self.linear}(\text{combined\_output})$$
$$\text{out} = \text{F.softmax}(\text{out}, \text{dim} = 2)$$

- **Output Layer:** The output layer generates the final predictions, representing the probabilities of the target word belonging to each possible sense.

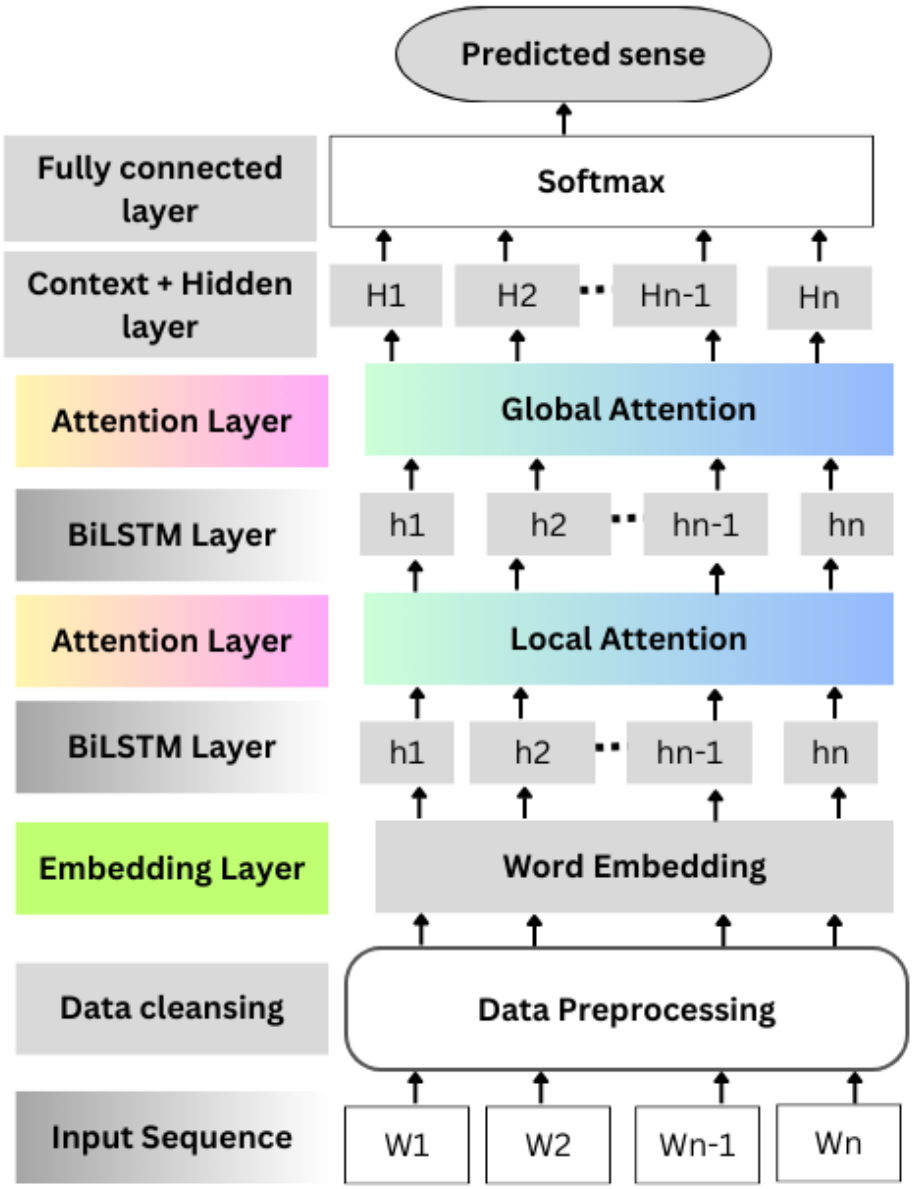


Figure 3.3: Proposed model architecture

We introduced our model architecture in Figure 3.3. Now, in the following figure, we break down the model into an encoder-decoder setup, showcasing detailed embedding for Amharic and Italian languages. To adapt our model, we employed various BERT variants for language-specific embeddings. The first encoder em-

ploys an attentive BiLSTM with local attention, receiving word embeddings and passing its context output to the second stacked BiLSTM, as illustrated in Figure 3.4. In the decoder, as shown in figure 3.5, the model utilizes a second BiLSTM layer followed by global attention and a softmax layer to generate a probability distribution for word senses based on the integrated context.

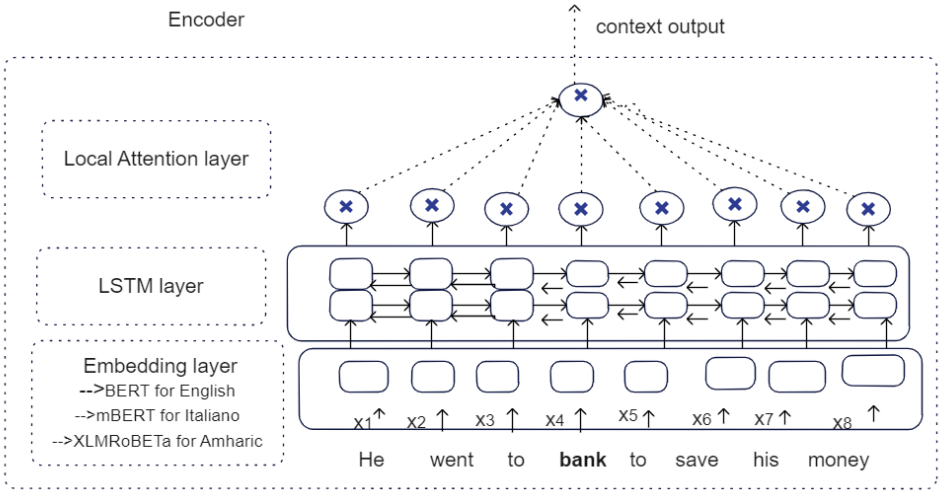


Figure 3.4: Encoder model

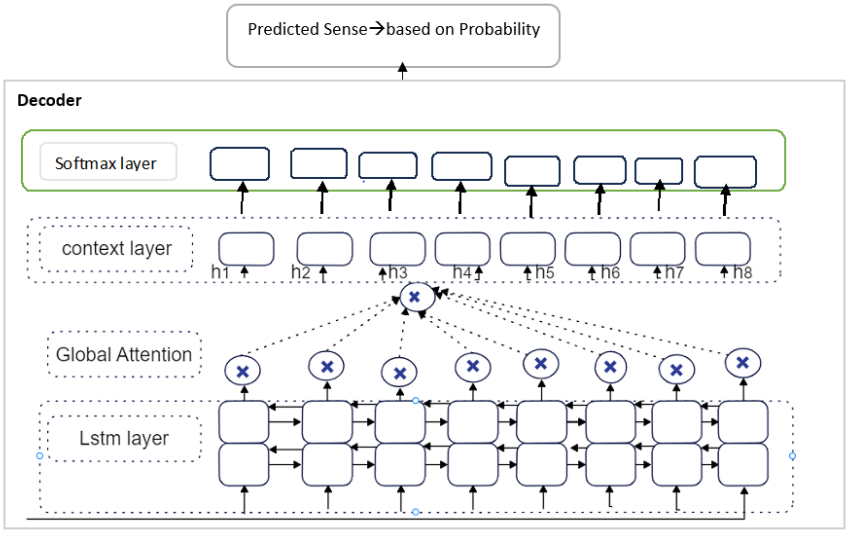


Figure 3.5: Decoder model

3.7 Evaluation Metrics

To assess the efficacy of the proposed approach for improving WSD, various essential evaluation metrics are utilized as stated below.

- **Accuracy:**
  - Accuracy quantifies the proportion of correctly disambiguated words out of the total number of words in the dataset.

$$\text{Accuracy} = \frac{\text{Number of correctly disambiguated words}}{\text{Total number of words in the dataset}}$$

- **Precision:**
  - Precision denotes the ratio of correctly predicted instances of a specific sense to the total predicted instances of that sense.

$$\text{Precision} = \frac{\text{Number of correctly predicted instances of a specific sense}}{\text{Total predicted instances of that sense}}$$

- **Recall:**
  - Recall, also referred to as sensitivity, represents the ratio of correctly predicted instances of a specific sense to the total actual instances of that sense.

$$\text{Recall} = \frac{\text{Number of correctly predicted instances of a specific sense}}{\text{Total actual instances of that sense}}$$

- **F1-Score:**
  - The F1-score is the harmonic mean of precision and recall, providing a balanced assessment of the two metrics.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Chapter 4

Experimentation

4.1 Introduction

This chapter focuses on providing detailed information about the experimental setup, including dataset selection, parameter configuration, and evaluation metrics. We conduct a systematic comparison between our proposed methodology and baseline methods to determine its efficacy in accurately disambiguating word senses.

4.2 Experimental Setup

The experimental setup contains various components, such as dataset selection and preprocessing, experimental configuration design, and identification of baseline methods for comparative analysis and evaluation. These elements are crucial in conducting a comprehensive evaluation of the proposed methodology.

Datasets

In our experimental evaluation, we chose two well-established benchmark datasets commonly used in word sense disambiguation research: WordNet and SemCor.

- **WordNet** We use WordNet to construct sense-tagged corpora and map synsets of words, which serve as training and testing data for our models.
- **SemCor**, a widely recognized training dataset for word sense disambiguation, was employed in our research as well.

Table 4.1: Training Datasets

Specification	Description
SemCor	Lexical corpus with sense annotations.
WSD Task Type	All-words WSD.
Dataset Link	https://www.sketchengine.eu/semcor-annotated-corpus/
Size	352 documents, 220,000 words, 37,000 sense labels
Data Split	80% Training / 10% Validation / 10% Testing

Table 4.2: Testing Datasets

Specification	Description
Senseval-2	English Lexical Sample [50]
Senseval-3	English Lexical Sample [51].
Senseval-2013	English Lexical Sample [48].
Senseval-2015	English All-words [49].
Size	1022 to 2282 sentences smallest to largest
Data Split	Training/Testing provided.

In our experimental setup, we used the following hardware and software. These specifications were chosen to ensure optimal performance and compatibility for our experiments in Word Sense Disambiguation.

- **Hardware:**
  - Central Processing Unit (CPU): Intel Core i7-7200U
  - Graphics Processing Unit (GPU): GPU Memory(google colab)
  - Memory: 12GB RAM
- **Software:**
  - Programming Language: Python 3.8
  - Libraries/Frameworks: PyTorch==1.10
  - Operating System: Windows 10

Table 4.3: Hyperparameters and Ranges

Hyperparameter	Range Selected
Learning rate	0.01 / 0.02 / **0.001**
Local context size	3
Epoch	32 / **50** / 100
Batch size	32 / **64**
No. of LSTM layers	2
Type of LSTM layer	Bidirectional
Dropout	0.1 / **0.2**
Word embedding size	100 / 200 / **300**
Initialization of scalar weights on Attentions	Random uniform (-0.1, 0.1)

In the evaluation, we employed specific hyper-parameter settings for our architectures, as outlined in Table 4.3. Our models were trained using the Adam optimizer, and the subsequent results reported in the next section are based on the use of Adam, as it yielded comparatively favorable outcomes. Our models were implemented using PyTorch 1.1 within a Windows environment.

We divided the data into training, testing, and validation sets. The ratios used were 80% for training, 10% for testing, and 10% for validation. The hyper-parameters were tuned exclusively using the validation set.

4.2.1 Training

For our development set, we utilize the widely recognized Semcor [105] dataset. All models in this study were trained for 32, 50, and 100 epochs with learning rates ranging from 0.001 to 0.1, using batch sizes of 32 and 64, as detailed in Table 4.3.

For Enhanced WSD, we utilized the **bert-base-uncased** model to obtain contextual representation of input sentences through contextual word embedding. Our **embedding layer** is implemented outside of the neural network and stored for use at any time. As an additional experiment, we tried using GloVe for word

embedding to compare its performance with the bert-base-uncased embedding. We found that bert-base-uncased is more dynamic and contextually aware than GloVe. Unlike GloVe, which provides static representations of words and ignores multiple senses within a sentence, bert-base-uncased offers more dynamic and context-aware embeddings.

We performed a series of experiments using different configurations.

- Initially, we implemented a basic BiLSTM model without an attention mechanism and BERT embeddings.
- Next, we introduced BERT embeddings into the model to enhance its performance.
- We then incorporated a local attention mechanism at the word level. The local context size was set to 3, allowing the model to focus on nearby words during processing.
- To further improve the model's performance, we implemented a global attention mechanism using Bahdanau attention. This approach helps overcome the limitations observed in traditional encoder-decoder architectures by alleviating the bottleneck problem.
- Finally, we combined the local and global attention mechanisms with the BERT embedding. This involved incorporating the output of the BERT embedding into the first BiLSTM layer of the model. By doing so, we aimed to further enhance the capabilities of our model.

To obtain the probability of the predicted sense, we utilize the **softmax** function on the final hidden state, which applies to all classes within the output vocabulary, encompassing the sense key vocabulary. To assess the robustness of our model, we adapted and tested our custom Amharic dataset.

4.3 Result Analysis

We present our experimental result and compare it with baseline works on the same framework. The performance results of our model, when compared to the latest advancements in the field as well as a basic baseline, are detailed in **Table** 4.4. As a supervised method, We evaluate our method against existing neural Word Sense Disambiguation models that utilize bidirectional Long Short-Term Memory networks by Raganato et al., 2017b [42]. They use a BiLSTM model with attention to WSD, jointly trained with a lexical semantic labeling task. This means the model learns to disambiguate word senses while also recognizing the semantic labels of words. Their approach aims to leverage the contextual information in sentences through the BiLSTM and attention mechanism, with its effectiveness potentially depending on the complexity and diversity of the training data. Given its solid foundation in the field, this work stands as a robust benchmark for many researchers.

We also compare our approach to previously established methods that incorporate gloss representation information from WordNet, contextual models like ELMo by M. E. Peter et al. [83] and BERT [36].

As we proceed with our analysis, comparing our results to various knowledge-based and supervised methods outlined in Table 4.4, we encounter **MFS** baseline approach, which entails selecting the **most frequent sense** based on the training data. However, it faces limitations in terms of generalizability and its inability to handle unseen data. The other work **Leskext+emb**, introduced by Basile et al. [148]. It takes inspiration from classic lesk algorithm (Lesk, 1986[13]). They employ a word similarity function in a distributional semantic space to assess the alignment between glosses and their contextual usage. This strategy, aimed at augmenting gloss information through semantic relationships, effectively enhances the performance of WSD tasks.

The other knowledge base system we consider is UKB by Aggrie et al. [149], and **Babelfy**, developed by Moro et al.[67], utilizes the semantic network structure of BabelNet to construct a cohesive graph-based framework for both Word Sense Disambiguation and Entity Linking tasks. All these methods are based on **MFS** baseline, which relies on most frequent sense.

Then, the development of the Supervised method improves and addresses the problem encountered by the above knowledge-based approaches. The paper related to our work: **Context2Vec** by Melamud et al., 2016[80] and **IMS**, introduced by Zhi and Ng (2010) [81], employs a linear SVM as its classifier. It utilizes a range of features within a limited window around the target word, including POS tags, nearby words, and local collocations. An extension of IMS, **IMS+emb**, as developed by Iacobacci et al. [82], retains IMS as its core framework but incorporates word embeddings as features. This adaptation has proven highly effective, often outperforming other methods across various WSD datasets.

As closest to our model which was developed using a neural network, by employing BiLSTM model. **Seq2Seq** model by M. Ahmed in 2018 [43], with extensive experiment **Seq2Seq + att.**, **Seq2Seq + att. + LEX**, **Seq2Seq + att. + LEX + POS**. Kaageback and Salomonsson developed **BiLSTM** model for WSD in 2016 [145]. When we begin implementing our model, start with simple BiLSTM as well. **Bi-LSTM+att.+LEX** and its modification **Bi-LSTM+att.+LEX+POS**, introduced by Raganato et al. 2017a[42], transform WSD into a sequence learning problem, are viewed as the closest baselines in our research. They introduce a multi-task learning setup that simultaneously tackles WSD, POS tagging, and coarse-grained semantic labels (LEX).

Overall, **Seq2Seq** and **BiLSTM** models outperform all of the other models. Introducing POS doesn't seem helpful as supported by previous work [167, 45].

To enhance both syntactic and semantic understanding, we implemented a BiLSTM model with a hierarchical attention mechanism and BERT embeddings.

This approach significantly improved performance, achieving state-of-the-art results across all WSD tasks, as demonstrated in the table below.

We also apply different techniques like **pos tagging**, **attention mechanism(both local and Global attention)**, hypernym compression, and **Bert embedding** separately and together to see the impact of different linguistic features to improve contextual understanding of the WSD model. As a result, employing local attention and global attention has an interesting performance improvement, also BERT embedding has a greater impact as we discussed in the evaluation section 4.3.1.

	Baseline papers			
Model	Senseval-2	Senseval-3	Semeval-2013	Semeval-2015
MFS	65.6	66.0	63.8	67.1
Leskext+emb[148]	63.0	63.7	66.2	64.6
UKBgloss w2w [149]	63.5	55.4	62.9	63.3
Babelfy[67]	67.0	63.5	66.4	70.3
IMS [81]	70.9	69.3	65.3	69.5
IMS+emb [82]	72.2	70.4	65.9	71.5
Context2Vec [80]	71.8	69.1	65.6	71.9
Seq2Seq [43]	68.5	67.9	65.3	67.0
Seq2Seq + att. [43]	69.9	69.6	65.6	67.7
Seq2Seq + att. + LEX [43]	70.6	67.8	6 6.5	68.7
Seq2Seq + att. + LEX + POS	70.1	68.5	66.5	69.2
Bi-LSTM_att+LEX+POS [42]	66.9	69.1	71.5	72.0
BiLSTM_att+LEX [42]	72.0	69.4	66.4	72.4
BERT	73.8	71.6	69.2	74.4
	Our Model			
BiLSTM_local_att.	66.2	67.4	69.3	69.6
BiLSTM_Global_att.	67.7	63.1	65.5	69.8
BiLSTM_embed+g. att.+l.att.	**74.2**	**72.3**	70.0	**75.2**

Table 4.4: Comparision of our model with Baseline papers for Word Sense Disambiguation with their evaluation in F1-score on various datasets. g.att. and l.att. indicates global attention and local attention respectively.

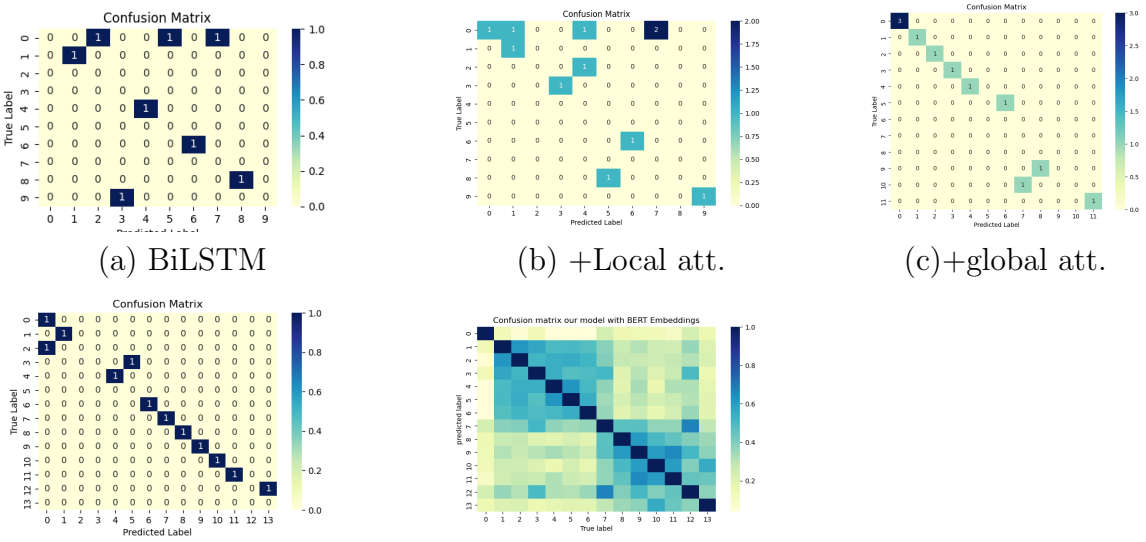
4.3.1 Evaluation

We conducted evaluations on all commonly used evaluation corpora for Word Sense Disambiguation, specifically the WSD tasks from the SensEval/SemEval evaluation dataset. Our evaluations utilized the corpora provided for this task by Raganato et al. 2017a [42], which included SensEval 2 (Edmonds and Cotton, 2001) [50], SensEval 3 (Snyder and Palmer, 2004) [51], SemEval 2013 [48], and SemEval 2015 task 13 (Moro and Navigli, 2015) [49].

In evaluating our model and comparing it with baselines and other models, we utilize the F1 score as a metric for smooth and standardized comparison. Although it may not be the most suitable metric for this task, the F1 score provides a balanced measure of precision and recall. By considering both false positives and false negatives, we gain insights into the model's accuracy and robustness. This approach allows us to make informed decisions and draw meaningful comparisons with other approaches in the field.

During testing, our models predict the most likely output words for a given target word by generating a probability distribution. This distribution is created using a Softmax layer at each step, which assigns probabilities to each potential class. The model ranks the candidate senses of the target word based on these probabilities, selecting the top-ranked candidates as its final output.

In our ablation study for Word Sense Disambiguation, we conducted a comparative analysis of various model configurations, each representing a different level of attention mechanism integration. The models were labeled as follows: BiLSTM **(Simple BiLSTM)** model, +Local att. **(BiLSTM with local attention)**, +global att. **(BiLSTM with global attention)**, +local & global att. **(BiLSTM with both local and global attention)**, and +embed.+local &global att. **(BiLSTM with both local and global attention + BERT embeddings)** respectively as indicated in figure 4.1 their prediction performance.



(a) BiLSTM (b) +Local att. (c)+global att.

(e) +local & global att. (f) +embed.+local &global att.

Figure 4.1: Confusion matrices for different models

Our initial findings revealed that the **BiLSTM** model performed reasonably well, although a noticeable number of classification errors were observed. Subsequently, incorporating local attention into the BiLSTM architecture resulted in a modest improvement of approximately **+1.3%** in accuracy. Moving forward, when we introduced global attention to the BiLSTM model, the impact on performance was not as pronounced as expected. However, when using BiLSTM with local attention, we encountered difficulties in handling longer sentences, particularly in capturing the context around the target word.

To address this challenge, we combined both attention mechanisms, shifting from dot-scale attention to Bahdanau attention for the global level **(sentence-level)** and employing multi-head attention for the local level **(word-level)** with a context size of **3** (although we also experimented with a context size of **5**). Remarkably, this adjustment yielded a substantial increase in accuracy of approximately **+6%** compared to the previous experiment, showcasing the effectiveness of integrating both local and global attention mechanisms in enhancing WSD performance.

Figure 4.2 presents the F1 scores (%) achieved by the baseline model, the

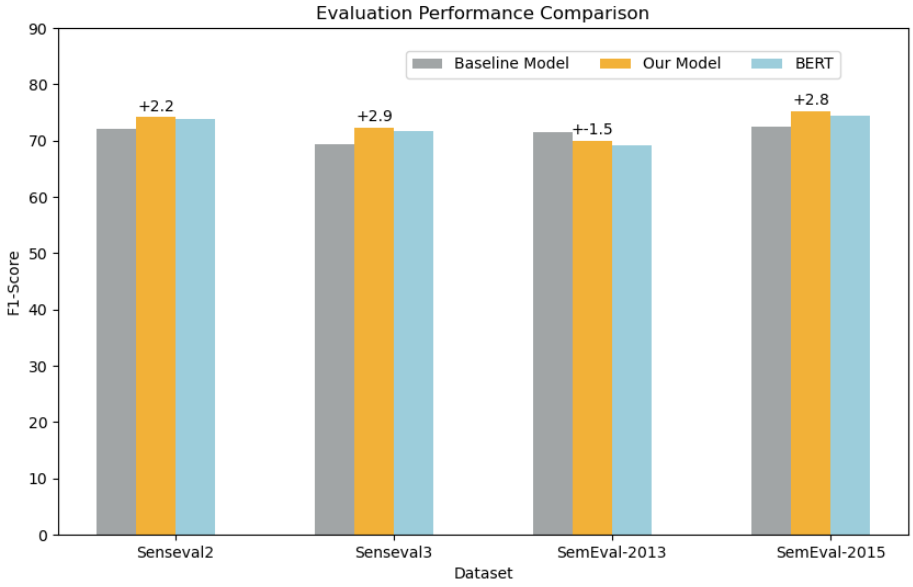
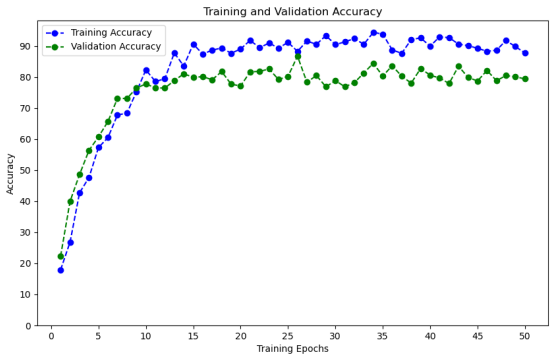


Figure 4.2: Proposed model Evaluation Performance Comparision

proposed model, and BERT on four different datasets used for word sense disambiguation: Senseval2, Senseval3, SemEval-2013, and SemEval-2015. The results demonstrate the performance improvement of the proposed model over BERT and the baseline model across these datasets. Specifically, Our Model shows an improvement in F1 score ranging from 1.8% to 2.9% compared to the baseline model.

(a) Simple BiLSTM



(b) Our Final model

Table 4.5: Our Model Performance on Training across epoch

The above figure 4.5 compares the training and validation performance plot of our initial model figure **(a)**, before incorporating BERT embedding and local

attention+global attention, and figure **(b)** with our final model, which includes these enhancements. The comparison demonstrates the improvement in accuracy achieved by incorporating BERT embedding and local attention+global attention in our model.

4.3.2 Adapting proposed model to other languages

We conducted a retraining process using mBERT embedding for the Italian language. We then compared our performance with the state-of-the-art results of a previous paper. Our comparisons were made against SensEmBERT, the best-performing all-part-of-speech system, and our baseline model, EWISER. The results are reported, including those for the Italian and low-resource language Amharic.

We presented the results in Table 4.6, which included evaluations on Semeval2013 (limited to nouns) and Semeval2015 (all parts of speech). The results indicated that our model outperforms the state-of-the-art papers for the italic language.

Table 4.6: Evaluation of WSD in Italian language reported in F1.

Model	SE-2013		SE-2015
	Train	Test	Test
Scozzafava et al.[168]	76.4	72.1	69.0
EWISER(Baseline) [44]	80.9	77.7	71.8
Ours	82.1	78.4	72.3

We could not compare our results for Amharic due to the lack of an available evaluation dataset. However, we prepared our dataset for this language. Despite the absence of a comparison, we observed significant improvements over the baseline systems, demonstrating the applicability of our model in multiple languages.

Table 4.7: Evaluation on Amharic WSD reported in F1.

Model	Own dataset	
	Train	Test
AmWSD(our)	92.4	78.2

4.4 Key Findings and Observations

Hierarchical Attention Mechanisms and BERT Embeddings: The study found that incorporating hierarchical attention mechanisms into BiLSTM models with BERT embeddings significantly improves WSD accuracy. This is particularly evident in complex sentences or for long-range dependencies, where the model's ability to capture both local and global context is crucial for accurate sense predictions. The model's ability to attend to contextual information beyond the ambiguous word, in conjunction with Bahdanau attention, shows promise in improving WSD accuracy by capturing broader context.

Enhanced Contextual Understanding: The proposed model's ability to understand contextual information through both local and global attention mechanisms leads to improved WSD performance. Local attention allows the model to focus on the context around the target word, capturing subtle nuances that aid in disambiguation. Global attention, on the other hand, enables the model to consider the entire sentence, providing a broader context that can influence sense predictions.

Adapting to other languages: The study demonstrates the effectiveness of the proposed model to improve WSD performance in Amharic and Italian languages. We retrain for both languages with our proposed model, by employing a multilingual bert and XLMRoberTa for Italian and Amharic respectively. We showed our model performs better over the baseline model for Italian language. Our model has been successfully adapted for Amharic and demonstrates strong performance.

Impact of Linguistic Features: The inclusion of Part-of-Speech (POS) tagging showed mixed results, indicating that while it can provide useful information, its impact on WSD performance may vary depending on the dataset and language. This suggests that careful consideration of linguistic features is necessary in developing effective WSD models.

Overall, the study's findings emphasize the importance of attention mechanisms, contextual embeddings, and linguistic features in developing accurate and effective WSD models. The proposed model's architecture, incorporating hierarchical attention mechanisms and BERT embeddings, shows significant advancements in WSD accuracy, particularly in challenging linguistic contexts.

Through a series of experiments and evaluations, we have addressed the following research questions:

1. **How does incorporating hierarchical attention mechanisms into BiLSTM models with BERT embeddings impact WSD accuracy, particularly in complex sentences or for long-range dependencies?**

   ▶ This integration significantly enhances WSD accuracy, particularly in complex sentences or for long-range dependencies. Hierarchical attention empowers the model to address these challenges by focusing on both local and global contexts, as well as analyzing specific words. BERT embeddings further contribute by providing rich contextual information about each word, enabling the model to make more accurate sense distinctions.

   - **Overall Accuracy Improvement**: Our model achieved significant improvements in overall accuracy, as evidenced by F1 score increases of 6.5% on the Senseval-2 dataset, 4.9% on Senseval-3, 0.7% on Senseval-13, and 5.6% on Senseval-2015 compared to our simple BiLSTM model. These enhancements can be attributed to the incorporation of our hierarchical attention mechanism and semantic integration.

- **Long-Range Dependencies**: For tasks requiring comprehension of long-range dependencies, hierarchical attention empowers the model to effectively focus on both local and global contexts, leading to better disambiguation of words with long-range dependencies.

2. **To what extent can the proposed model's enhanced ability to understand contextual information through both local and global attention mechanisms lead to improved word sense disambiguation?**

   ▶ The improved capability of the proposed model to comprehend contextual information through local and global attention mechanisms greatly enhances Word Sense Disambiguation.

   - **F1 score Improvement**: Our proposed model achieves significant improvements in F1 score over baseline models on four standard benchmark datasets. The improvements range from 2.2% on Senseval-2 to 2.8% on SemEval-2015, as evidenced by our testing. Additionally, compared to a BERT-based model for word sense disambiguation, our model achieves further gains of 0.4% to 0.8% on the same datasets (Senseval-2, Senseval-3, SemEval-2013, and SemEval-2015) respectively.
   - Local vs. Global Attention: Local attention helps the model focus on the immediate context, capturing nuances in nearby words, while global attention encompasses the entire sentence. This dual focus leads to a more holistic understanding, significantly improving disambiguation accuracy.

3. **Can the proposed model be adapted to improve performance for Amharic word sense disambiguation?**

▶ Yes, Our model has been successfully adapted for Amharic and demonstrates strong performance. By leveraging pre-trained models and embeddings from XLMAmRoberta which is a variant of BERT fine-tuned for the Amharic language, the model effectively adapted to the Amharic word sense disambiguation. We have achieved 92.4% F1-score on training data and 78.2% F1-score on test data.

Chapter 5

Concluding Remarks

5.1 Conclusion

In this thesis, we demonstrated how hierarchical attention and semantic integration can effectively address semantic word sense ambiguity. Our approach, which enhances Word Sense Disambiguation accuracy using hierarchical attention mechanisms and BERT embeddings, incorporates local and global attention to leverage the rich contextual information provided by BERT embeddings. Our results show a notable enhancement in WSD accuracy, especially in complex sentences and for long-range dependencies. In addition to our methodological contributions, our study revealed several key insights. **Firstly**, hierarchical attention mechanisms proved effective in capturing pertinent context for accurate disambiguation. **Moreover**, our model exhibited enhanced performance in comprehending contextual information by utilizing both local and global attention. **Notably**, we also successfully adapted our model and improved performance in Amharic, a low-resource language. **Lastly**, our thesis highlights the importance of attention mechanisms and linguistic features, like Part-of-Speech (POS) tagging, in WSD models. While POS tagging showed almost less impact on the English dataset and a high impact on both the Italian and Amharic languages, attention mechanisms significantly improved accuracy by capturing broader context.

Overall, our research enhances WSD models by leveraging hierarchical attention and BERT embeddings, improving accuracy in complex linguistic contexts, and adapting our model to Italian and Amharic languages.

5.2 Future directions

- **Integration into Downstream Applications**

  We plan to leverage the flexibility of our models by integrating them into downstream applications, such as Machine Translation (MT) and Information Retrieval(IR).

- **Extended model development:** As a recommendation for future work, developing domain-specific WSD models for specialized domains such as biomedical texts, legal documents, and technical literature would be beneficial. By focusing on domain-adaptive WSD models, researchers can address the unique challenges posed by specialized terminology and discourse in these domains. **Furthermore**, research on multimodal WSD, integrating visual and auditory context alongside textual information, could lead to improved sense disambiguation in multimedia-rich environments.

- **Extended Evaluation and Adaptation**

  Larger Corpora Evaluation: Extend evaluation to larger corpora to validate the scalability and robustness of the proposed model across diverse datasets, ensuring its effectiveness in real-world applications.

  Pretrained Model Exploration: Conduct an extensive exploration of pretrained models, such as BERT, GPT, or XLNet, to identify optimal configurations and further enhance the performance of the proposed WSD model.

In summary, it is crucial for future research to prioritize the development of resilient methodologies, delve into innovative approaches, and tackle real-world applications. These efforts will foster innovation and make a tangible impact in the field of Word Sense Disambiguation.

REFERENCES

[1] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, and X. Zhou, "Semantics-aware bert for language understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9628–9635.

[2] L. Chen, P. Chen, and Z. Lin, "Artificial intelligence in education: A review," *Ieee Access*, vol. 8, pp. 75 264–75 278, 2020.

[3] E. Pelivani and B. Cico, "Toward self-aware machines: Insights of causal reasoning in artificial intelligence," in *2021 International Conference on Information Technologies (InfoTech)*. IEEE, 2021, pp. 1–4.

[4] J. H. Korteling, G. C. van de Boer-Visschedijk, R. A. Blankendaal, R. C. Boonekamp, and A. R. Eikelboom, "Human-versus artificial intelligence," *Frontiers in artificial intelligence*, vol. 4, p. 622364, 2021.

[5] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia tools and applications*, vol. 82, no. 3, pp. 3713–3744, 2023.

[6] R. Navigli, "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009.

[7] V. Iyer, P. Chen, and A. Birch, "Towards effective disambiguation for machine translation with large language models," *arXiv preprint arXiv:2309.11668*, 2023.

[8] D. Vickrey, L. Biewald, M. Teyssier, and D. Koller, "Word-sense disambiguation for machine translation," in *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 2005, pp. 771–778.

[9] J. Gomes Jr, R. C. de Mello, V. Ströele, and J. F. de Souza, "A study of approaches to answering complex questions over knowledge bases," *Knowledge and Information Systems*, vol. 64, no. 11, pp. 2849–2881, 2022.

[10] Z. Zhong and H. T. Ng, "Word sense disambiguation improves information retrieval," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 273–282.

[11] C. Hung and S.-J. Chen, "Word sense disambiguation based sentiment lexicons for sentiment classification," *Knowledge-Based Systems*, vol. 110, pp. 224–232, 2016.

[12] W. Weaver, "Information theory," *eM Publications*, p. 232, 1949.

[13] M. Lesk, "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*, 1986, pp. 24–26.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[16] A. Kilgarriff and M. Palmer, "Introduction to the special issue on senseval," *Computers and the Humanities*, vol. 34, pp. 1–13, 2000.

[17] P. Nanjundan and E. Z. Mathews, "An analysis of word sense disambiguation (wsd)," in *Proceedings of the International Health Informatics Conference: IHIC 2022*. Springer, 2023, pp. 251–259.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.

[20] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, "Detecting formal thought disorder by deep contextualized word representations," *Psychiatry Research*, vol. 304, p. 114135, 2021.

[21] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," *OpenAI*, 2018.

[22] P.-Y. Vandenbussche, T. Scerri, and R. Daniel Jr, "Word sense disambiguation with transformer models," in *Proceedings of the 6th Workshop on Semantic Deep Learning (SemDeep-6)*, 2021, pp. 7–12.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] D. Loureiro, K. Rezaee, M. T. Pilehvar, and J. Camacho-Collados, "Analysis and evaluation of language models for word sense disambiguation," *Computational Linguistics*, vol. 47, no. 2, pp. 387–443, 2021.

[25] N. Rahman and B. Borah, "Improvement of query-based text summarization using word sense disambiguation," *Complex & Intelligent Systems*, vol. 6, pp. 75–85, 2020.

[26] R. Jose and V. S. Chooralil, "Prediction of election result by enhanced sentiment analysis on twitter data using word sense disambiguation," in *2015 International Conference on Control Communication & Computing India (ICCC)*. IEEE, 2015, pp. 638–641.

[27] F. B. Mesmia, M. Mouhoub, D. Xie, F. Li, B. Li, C. Teng, D. Ji, M. Zhang, Y. Chen, W. Zhang *et al.*, "Asian and low-resource language information processing," *ACM Transactions on*, vol. 22, no. 11, 2023.

[28] D. Yarowsky and R. Florian, "Evaluating sense disambiguation across diverse parameter spaces," *Natural Language Engineering*, vol. 8, no. 4, pp. 293–310, 2002.

[29] W. A. Gale, K. Church, and D. Yarowsky, "Estimating upper and lower bounds on the performance of word-sense disambiguation programs," in *30th Annual Meeting of the Association for Computational Linguistics*, 1992, pp. 249–256.

[30] Y. Goldberg, *Neural network methods for natural language processing*. Switzerland: Springer Nature, 2022.

[31] D. Rothman, *Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more*. Birmingham,UK: Packt Publishing Ltd, 2021.

[32] Y. Sun and J. Platoš, "Attention-based stacked bidirectional long short-term memory model for word sense disambiguation," *Association for Computing Machinery*, 2023. [Online]. Available: https://doi.org/10.1145/3594780

[33] G. Tang, R. Sennrich, and J. Nivre, "An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation," in *Conference on Machine Translation*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:52984984

[34] L. Huang, C. Sun, X. Qiu, and X. Huang, "Glossbert: Bert for word sense disambiguation with gloss knowledge," *arXiv preprint arXiv:1908.07245*, 2019.

[35] D. A. B. Loureiro, "Learning word sense representations from neural language models," *Computational Linguistics*, 2023.

[36] C. Hadiwinoto, H. T. Ng, and W. C. Gan, "Improved word sense disambiguation using pre-trained contextualized word representations," *arXiv preprint arXiv:1910.00194*, 2019.

[37] L. Zhang, "Word sense disambiguation model based on bi-lstm," in *2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. IEEE, 2022, pp. 848–851.

[38] B. Mutlum, "Word sense disambiguation based on sense similarity and syntactic context," Ph.D. dissertation, Citeseer, 2005.

[39] N. Mossa and M. Meshesha, "Amharic sentence-level word sense disambiguation u sing transfer learning," in *Artificial Intelligence and Digitalization for Sustainable Development: 10th EAI International Conference, ICAST 2022, Bahir Dar, Ethiopia, November 4-6, 2022, Proceedings*. Springer, 2023, pp. 227–238.

[40] S. M. Dereje, T. Y. Tesfa, W. T. Yitbarek *et al.*, "Sentence level amharic word sense disambiguation," *American Journal of Education and Technology*, vol. 1, no. 2, pp. 83–87, 2022.

[41] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[42] A. Raganato, C. D. Bovi, and R. Navigli, "Neural sequence learning models for word sense disambiguation," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 1156–1167.

[43] M. Ahmed, M. R. Samee, and R. Mercer, "A novel neural sequence model with multiple attentions for word sense disambiguation," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 687–694.

[44] M. Bevilacqua, R. Navigli *et al.*, "Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 2854–2864.

[45] H. M. Alonso and B. Plank, "When is multitask learning effective? semantic sequence prediction under varying data conditions," *arXiv preprint arXiv:1612.02251*, 2016.

[46] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas, "Using a semantic concordance for sense identification," in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[47] K. C. Litkowski and O. Hargraves, "Semeval-2007 task 06: Word-sense disambiguation of prepositions," in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 2007, pp. 24–29.

[48] E. Lefever and V. Hoste, "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)*, 2013, pp. 158–166.

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

[50] P. Edmonds and S. Cotton, "Senseval-2: overview," in *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, 2001, pp. 1–5.

[51] R. Mihalcea, T. Chklovski, and A. Kilgarriff, "The senseval-3 english lexical sample task," in *Proceedings of SENSEVAL-3, the third international workshop on the evaluation of systems for the semantic analysis of text*, 2004, pp. 25–28.

[52] R. Mitkov, *The Oxford handbook of computational linguistics*. Oxford University Press, 2022.

[53] H. Singh and P. Bhattacharyya, "A survey on word sense disambiguation," *ACM Comput. Surv.(CSUR)*, 2019.

[54] R. Navigli, "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 2, pp. 1–69, 2009.

[55] R. Navigli and S. P. Ponzetto, "Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial intelligence*, vol. 193, pp. 217–250, 2012.

[56] D. S. Chaplot and R. Salakhutdinov, "Knowledge-based word sense disambiguation using topic models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[57] S. Olika, "Word sense disambiguation for afaan oromo: Using knowledge base," Ph.D. dissertation, St. Mary's University, 2018.

[58] W. Weaver, "Typescript, july 1949. translation. reprinted in locke, william n., and a. donald booth (eds) machine translation of languages: fourteen essays," 1955.

[59] M. Masterman, "The thesaurus in syntax and semantics." *Mech. Transl. Comput. Linguistics*, vol. 4, no. 1-2, pp. 35–43, 1957.

[60] S. Madhu and D. W. Lytle, "A figure of merit technique for the resolution of non-grammatical ambiguity." *Mech. Transl. Comput. Linguistics*, vol. 8, no. 2, pp. 9–13, 1965.

[61] K. Sparck Jones, *Synonymy and semantic classification.* Edinburgh University Press, 1986.

[62] R. Mante, M. Kshirsagar, and P. Chatur, "A review of literature on word sense disambiguation," *Int. J. Comput. Sci. Inf. Technol.(IJCSIT)*, vol. 5, no. 2, pp. 1475–1477, 2014.

[63] S. Banerjee, T. Pedersen *et al.*, "Extended gloss overlaps as a measure of semantic relatedness," in *Ijcai*, vol. 3, 2003, pp. 805–810.

[64] P. Basile, A. Caputo, and G. Semeraro, "An enhanced lesk word sense disambiguation algorithm through a distributional semantic model," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers.* Dublin, Ireland: Dublin City University and Association for Computational Linguistics, August 2014, pp. 1591–1600.

[65] E. Agirre, O. López de Lacalle, and A. Soroa, "Random walks for knowledge-based word sense disambiguation," *Computational Linguistics*, vol. 40, no. 1, pp. 57–84, 2014.

[66] A. Raganato, J. Camacho-Collados, R. Navigli *et al.*, "Word sense disambiguation: a uinified evaluation framework and empirical comparison," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, vol. 1, 2017, pp. 99–110.

[67] A. Moro, A. Raganato, and R. Navigli, "Entity linking meets word sense disambiguation: a unified approach," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 231–244, 2014.

[68] Y. Wang, M. Wang, and H. Fujita, "Word sense disambiguation: A comprehensive knowledge exploitation framework," *Knowledge-Based Systems*, vol. 190, p. 105030, 2020.

[69] M. AlMousa, R. Benlamri, and R. Khoury, "A novel word sense disambiguation approach using wordnet knowledge graph," *Computer Speech & Language*, vol. 74, p. 101337, 2022.

[70] M. Quillian, "Semantic memory. m. minsky, editor, semantic information processing," 1968.

[71] R. F. Simmons *et al.*, *Semantic networks: Their computation and use for understanding English sentences*. Department of Computer Sciences and Computer-Assisted Instruction Laboratory . . . , 1972.

[72] Y. Wilks, "A preferential, pattern-seeking, semantics for natural language inference," *Artificial intelligence*, vol. 6, no. 1, pp. 53–74, 1975.

[73] G. Hirst and E. Charniak, "Word sense and case slot disambiguation," in *Proceedings of the Second AAAI Conference on Artificial Intelligence*, 1982, pp. 95–98.

[74] G. Cottrell and J. Allen, "A connectionist approach to word sense disambiguation," Ph.D. dissertation, University of Porto, 1985.

[75] P. P. Borah, G. Talukdar, and A. Baruah, "Approaches for word sense disambiguation–a survey," *International Journal of Recent Technology and Engineering*, vol. 3, no. 1, pp. 35–38, 2014.

[76] E. Black, "An experiment in computational discrimination of english word senses," *IBM Journal of research and development*, vol. 32, no. 2, pp. 185–194, 1988.

[77] J. Sarmah and S. K. Sarma, "Decision tree based supervised word sense disambiguation for assamese," *Int. J. Comput. Appl*, vol. 141, no. 1, pp. 42–48, 2016.

[78] G. Escudero, L. Màrquez, and G. Rigau, "Naive bayes and exemplar-based approaches to word sense disambiguation revisited," *arXiv preprint cs/0007011*, 2000.

[79] Y. K. Lee and H. T. Ng, "An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 2002, pp. 41–48.

[80] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional lstm," in *Proceedings of the 20th*

SIGNLL conference on computational natural language learning, 2016, pp. 51–61.

[81] Z. Zhong and H. T. Ng, "It makes sense: A wide-coverage word sense disambiguation system for free text," in *Proceedings of the ACL 2010 system demonstrations*, 2010, pp. 78–83.

[82] I. J. Iacobacci, M. T. Pilehvar, R. Navigli *et al.*, "Embeddings for word sense disambiguation: An evaluation study," in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016-Long Papers*, vol. 2. Association for Computational Linguistics (ACL), 2016, pp. 897–907.

[83] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018.

[84] F. Luo, T. Liu, Q. Xia, B. Chang, and Z. Sui, "Incorporating glosses into neural word sense disambiguation," *arXiv preprint arXiv:1805.08028*, 2018.

[85] L. Vial, B. Lecouteux, and D. Schwab, "Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation," *arXiv preprint arXiv:1905.05677*, 2019.

[86] Y. Song, X. C. Ong, H. T. Ng, and Q. Lin, "Improved word sense disambiguation with enhanced sense representations," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 4311–4320.

[87] E. Barba, L. Procopio, R. Navigli *et al.*, "Consec: Word sense disambiguation as continuous sense comprehension," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 1492–1503.

[88] R. Chasin, A. Rumshisky, O. Uzuner, and P. Szolovits, "Word sense disambiguation in the clinical domain: a comparison of knowledge-rich and knowledge-poor unsupervised methods," *Journal of the American Medical Informatics Association*, vol. 21, no. 5, pp. 842–849, 2014.

[89] A. Aliwy and H. Taher, "Word sense disambiguation: Survey study," *Journal of Computer Science*, vol. 15, pp. 1004–1011, 07 2019.

[90] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, Jun. 1995, pp. 189–196. [Online]. Available: https://aclanthology.org/P95-1026

[91] A. Saeed, R. M. A. Nawab, and M. Stevenson, "Investigating the feasibility of deep learning methods for urdu word sense disambiguation,"

Association for Computing Machinery, vol. 21, no. 2, oct 2021. [Online]. Available: https://doi.org/10.1145/3477578

[92] H. Calvo, A. P. Rocha-Ramírez, M. A. Moreno-Armendáriz, and C. A. Duchanoy, "Toward universal word sense disambiguation using deep neural networks," *IEEE Access*, vol. 7, pp. 60 264–60 275, 2019.

[93] K.-H. Nguyen and C.-Y. Ock, "Margin perceptron for word sense disambiguation," in *Proceedings of the 1st Symposium on Information and Communication Technology*, 2010, pp. 64–70.

[94] V. Singh and P. Kumar, "Word sense disambiguation for punjabi language using deep learning techniques," *Neural Computing and Applications*, vol. 32, 04 2020.

[95] A. Popov, "Word sense disambiguation with recurrent neural networks," in *Proceedings of the Student Research Workshop associated with RANLP*, vol. 2017, 2017, pp. 25–34.

[96] S. Yang, H.-C. Chen, C.-H. Wu, M.-N. Wu, and C.-H. Yang, "Forecasting of the prevalence of dementia using the lstm neural network in taiwan," *Mathematics*, vol. 9, no. 5, 2021. [Online]. Available: https://www.mdpi.com/2227-7390/9/5/488

[97] A. Graves and A. Graves, *Supervised sequence labelling*. Springer, 2012.

[98] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[99] E. Lodhi, F.-Y. Wang, G. Xiong, L. Zhu, T. S. Tamir, W. U. Rehman, and M. A. Khan, "A novel deep stack-based ensemble learning approach for fault detection and classification in photovoltaic arrays," *Remote Sensing*, vol. 15, no. 5, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/5/1277

[100] J. H. Martin, "Language divergences and typology," in *2018 Oriental COCOSDA - International Conference on Speech Database and Assessments*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:231772629

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

[102] S. Olika, "Word sense disambiguation for afaan oromo: Using knowledge base," Ph.D. dissertation, St. Mary's University, 2018.

[103] H. Calvo, A. P. Rocha-Ramirez, M. A. Moreno-Armendáriz, and C. A. Duchanoy, "Toward universal word sense disambiguation using deep neural networks," *IEEE Access*, vol. 7, pp. 60 264–60 275, 2019.

[104] A. Breit, A. Revenko, K. Rezaee, M. T. Pilehvar, and J. Camacho-Collados, "Wic-tsv: An evaluation benchmark for target sense verification of words in context," *arXiv preprint arXiv:2004.15016*, 2020.

[105] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas, "Using a semantic concordance for sense identification," in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. [Online]. Available: https://aclanthology.org/H94-1046

[106] R. Navigli and S. P. Ponzetto, "Babelnet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012. [Online]. Available: https://madoc.bib.uni-mannheim.de/35171/

[107] G. R. Haffari and A. Sarkar, "Analysis of semi-supervised learning with the yarowsky algorithm," *arXiv preprint arXiv:1206.5240*, 2012.

[108] E. Agirre and P. Edmonds, *Word sense disambiguation: Algorithms and applications*. Oxford,UK: Springer Science & Business Media, 2007, vol. 33.

[109] M. Stevenson and Y. Wilks, "The interaction of knowledge sources in word sense disambiguation," *Computational Linguistics*, vol. 27, no. 3, pp. 321–349, 2001.

[110] A. R. Pal, A. Kundu, A. Singh, R. Shekhar, and K. Sinha, "A hybrid approach to word sense disambiguation combining supervised and unsupervised learning," *arXiv preprint arXiv:1611.01083*, 2015.

[111] S. Elmougy, H. Taher, and H. Noaman, "Naïve bayes classifier for arabic word sense disambiguation," in *proceeding of the 6th International Conference on Informatics and Systems*. Citeseer, 2008, pp. 16–21.

[112] Y. Luan, B. Hauer, L. Mou, and G. Kondrak, "Improving word sense disambiguation with translations," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4055–4065. [Online]. Available: https://aclanthology.org/2020.emnlp-main.332

[113] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[114] H. Kang, T. Blevins, and L. Zettlemoyer, "Translate to disambiguate: Zero-shot multilingual word sense disambiguation with pretrained language models," *arXiv preprint arXiv:2304.13803*, 2023.

[115] X.-R. Sun, S.-H. Lv, X.-D. Wang, and D. Wang, "Chinese word sense disambiguation using a lstm," in *ITM Web of Conferences*, vol. 12. EDP Sciences, 2017, p. 01027.

[116] Y. Z. and H. H., "Graph based word sense disambiguation method using distance between words," *Journal of Software*, vol. 23, no. 4, pp. 776–785, 2012.

[117] W. Lu, F. Meng, S. Wang, G. Zhang, X. Zhang, A. Ouyang, and X. Zhang, "Graph-based chinese word sense disambiguation with multi-knowledge integration." *Computers, Materials & Continua*, vol. 61, no. 1, 2019.

[118] B. Hou, F. Qi, Y. Zang, X. Zhang, Z. Liu, and M. Sun, "Try to substitute: An unsupervised chinese word sense disambiguation method based on hownet," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec 2020, pp. 1752–1757. [Online]. Available: https://aclanthology.org/2020.coling-main.155

[119] X. Zhang, B. Hauer, and G. Kondrak, "Improving hownet-based chinese word sense disambiguation with translations," in *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec 2022, pp. 4530–4536. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.331

[120] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for chinese bert," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, p. 3504–3514, Nov 2021. [Online]. Available: https://doi.org/10.1109/TASLP.2021.3124365

[121] F. Luo, T. Liu, Z. He, Q. Xia, Z. Sui, and B. Chang, "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*, 2018, pp. 1402–1411.

[122] L. Huang, C. Sun, X. Qiu, and X. Huang, "Glossbert: Bert for word sense disambiguation with gloss knowledge," in *Proceedings*

of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, Nov 2019, pp. 3509–3514. [Online]. Available: https://aclanthology.org/D19-1355

[123] B. Elayeb, "Arabic word sense disambiguation: a review," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2475–2532, 2019.

[124] N. Habash and O. Rambow, "Arabic diacritization through full morphological tagging," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, 2007, pp. 53–56.

[125] I. Zitouni, *Natural language processing of semitic languages*. New York,USA: Springer, 2014.

[126] F. Debili, H. Achour, and E. Souissi, "La langue arabe et l'ordinateur: de l'étiquetage grammatical à la voyellation automatique," *Correspondances*, vol. 71, pp. 10–28, 2002.

[127] A. Chalabi, "Sakhr arabic-english computer-aided translation system," in *Conference of the Association for Machine Translation in the Americas*. Springer, 1998, pp. 518–521.

[128] E. Atwell, L. Al-Sulaiti, S. Al-Osaimi, and B. Abu Shawar, "A review of arabic corpus analysis tools," in *Proceedings of TALN04: XI conference sur le traitement automatique des langues naturelles*, vol. 2, 2004, pp. 229–234.

[129] I. Bounhas, B. Elayeb, F. Evrard, and Y. Slimani, "Arabonto: experimenting a new distributional approach for building arabic ontological resources," *International Journal of Metadata, Semantics and Ontologies*, vol. 6, no. 2, pp. 81–95, 2011.

[130] ——, "Information reliability evaluation: from arabic storytelling to computer sciences," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 8, no. 3, pp. 1–33, 2015.

[131] N. Habash, R. Roth, O. Rambow, R. Eskander, and N. Tomeh, "Morphological analysis and disambiguation for dialectal arabic," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 426–432.

[132] M. S. Priya, D. K. Renuka, L. A. Kumar, and S. L. Rose, "Multilingual low resource indian language speech recognition and spell correction using indic bert," *Sādhanā*, vol. 47, no. 4, p. 227, 2022.

[133] A. R. Pal and D. Saha, "Word sense disambiguation: A survey," *arXiv preprint arXiv:1508.01346*, 2015.

[134] U. Gawande, S. Kale, and C. Thaokar, "A novel approach of word sense disambiguation for marathi language using machine learning," in *Recent Advances in Material, Manufacturing, and Machine Learning*. Abingdon, Oxon, OX14 4RN: CRC Press, 2023, pp. 643–652.

[135] R. L. Singh, K. Ghosh, K. Nongmeikapam, and S. Bandyopadhyay, "A decision tree based word sense disambiguation system in manipuri language," *Advanced Computing*, vol. 5, no. 4, p. 17, 2014.

[136] S. Gopal and R. P. Haroon, "Malayalam word sense disambiguation using naïve bayes classifier," in *2016 International Conference on Advances in Human Machine Interaction (HMI)*, 2016, pp. 1–4.

[137] S. Mekonen, "Word sense disambiguation for amharic text: A machine learning approach," *Unpublished Master's Thesis*, pp. 1–94, 2010.

[138] M. Mulugeta, "Word sense disambiguation for amharic sentences using wordnet hierarchy," Ph.D. dissertation, Bahirdar University, 2020.

[139] G. Wassie, B. Ramesh, S. Teferra, and M. Meshesha, "A word sense disambiguation model for amharic words using semi-supervised learning paradigm," *Science, Technology and Arts Research Journal*, vol. 3, no. 3, pp. 147–155, 2014.

[140] T. Kassie, "Word sense disambiguation for amharic text retrieval: A case study for legal documents," *Addis Ababa, Ethiopia. Masters Thesis Addis Ababa University, Ethiopia*, 2009.

[141] W. Hagerie, "Ensemble classifiers applied to amharic word sense disambiguation," *Addis Ababa University*, 2013.

[142] N. G. Kharate and V. H. Patil, "Word sense disambiguation for marathi language using wordnet and the lesk approach," in *Proceeding of First Doctoral Symposium on Natural Computing Research: DSNCR 2020*. Springer, 2021, pp. 45–54.

[143] S. Assemu, "Unsupervised machine learning approach for word sense disambiguation to amharic words," *Unpublished Master's Thesis, Department of Information Science, Addis Ababa University, Addis Ababa, Ethiopia*, 2011.

[144] A. Popov, "Neural network models for word sense disambiguation: an overview," *Cybernetics and information technologies*, vol. 18, no. 1, pp. 139–151, 2018.

[145] M. Kågebäck and H. Salomonsson, "Word sense disambiguation using a bidirectional lstm," *arXiv preprint arXiv:1606.03568*, 2016.

[146] C. Zhang, D. Biś, X. Liu, and Z. He, "Biomedical word sense disambiguation with bidirectional long short-term memory and attention-based neural networks," *BMC bioinformatics*, vol. 20, pp. 1–15, 2019.

[147] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[148] P. Basile, A. Caputo, and G. Semeraro, "An enhanced lesk word sense disambiguation algorithm through a distributional semantic model," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 1591–1600.

[149] E. Agirre, O. L. de Lacalle, and A. Soroa, "The risk of sub-optimal use of open source nlp software: Ukb is inadvertently state-of-the-art in knowledge-based wsd," *arXiv preprint arXiv:1805.04277*, 2018.

[150] C.-X. Zhang, Y.-L. Zhang, and X.-Y. Gao, "Multi-head self-attention gated-dilated convolutional neural network for word sense disambiguation," *IEEE Access*, vol. 11, pp. 14 202–14 210, 2023.

[151] J. Cheng, W. Tong, and W. Yan, "Capsule network improved multi-head attention for word sense disambiguation," *Applied Sciences*, vol. 11, no. 6, p. 2488, 2021.

[152] T. Domhan, "How much attention do you need? a granular analysis of neural machine translation architectures," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1799–1808.

[153] M. Ahmed, M. R. Samee, and R. Mercer, "A novel neural sequence model with multiple attentions for word sense disambiguation," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 687–694.

[154] G. Tang, R. Sennrich, and J. Nivre, "An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, C. Monz, M. Negri, A. Névéol, M. Neves, M. Post, L. Specia, M. Turchi, and K. Verspoor, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 26–35. [Online]. Available: https://aclanthology.org/W18-6304

[155] M. Y. Kang, T. H. Min, and J. S. Lee, "Sense space for word sense disambiguation," in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2018, pp. 669–672.

[156] K. Orkphol and W. Yang, "Word sense disambiguation using cosine similarity collaborates with word2vec and wordnet," *Future Internet*, vol. 11, no. 5, p. 114, 2019.

[157] A. Kutuzov and E. Kuzmenko, "To lemmatize or not to lemmatize: how word normalisation affects elmo performance in word sense disambiguation," *arXiv preprint arXiv:1909.03135*, 2019.

[158] G. Wiedemann, S. Remus, A. Chawla, and C. Biemann, "Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings," *arXiv preprint arXiv:1909.10430*, 2019.

[159] B. Scarlini, T. Pasini, and R. Navigli, "Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8758–8765.

[160] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[161] S. Zobaed, M. E. Haque, M. F. Rabby, and M. A. Salehi, "Senspick: Sense picking for word sense disambiguation," in *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, 2021, pp. 318–324.

[162] G. Tang, R. Sennrich, and J. Nivre, "An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation," *arXiv preprint arXiv:1810.07595*, 2018.

[163] K. Peffers, T. Tuunanen, C. E. Gengler, M. Rossi, W. Hui, V. Virtanen, and J. Bragge, "Design science research process: A model for producing and presenting information systems research," *arXiv preprint arXiv:2006.02763*, 2020.

[164] C. P. Chai, "Comparison of text preprocessing methods," *Natural Language Engineering*, vol. 29, no. 3, pp. 509–553, 2023.

[165] M. Gasser, "Hornmorpho: a system for morphological processing of amharic, oromo, and tigrinya," in *Conference on Human Language Technology for Development, Alexandria, Egypt*, 2011, pp. 94–99.

[166] T. D. Belay, A. A. Ayele, G. Gelaye, S. M. Yimam, and C. Biemann, "Impacts of homophone normalization on semantic models for amharic," in

2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA). IEEE, 2021, pp. 101–106.

[167] A. Søgaard and Y. Goldberg, "Deep multi-task learning with low level tasks supervised at lower layers," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 231–235.

[168] F. Scozzafava, M. Maru, F. Brignone, G. Torrisi, R. Navigli *et al.*, "Personalized pagerank with syntagmatic information for multilingual word sense disambiguation," in *Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations*. Association for Computational Linguistics, 2020, pp. 37–46.