COMPARATIVE ANALYSIS OF TRANSFORMER-BASED ARCHITECTURES AND STRATEGIES FOR WORD SENSE DISAMBIGUATION

Artur Farrakhov
Czech Technical University in Prague
Faculty of Information Technology
farraart@cvut.cz

ABSTRACT

Word Sense Disambiguation (WSD) remains a fundamental challenge in Natural Language Processing, requiring methods that can recognize subtle semantic nuances in varying contexts. This paper presents a comparative study of encoder-only Transformer architectures, including BERT, RoBERTa, and DeBERTa, applied to the WSD task. Beyond standard token classification, the research investigates advanced strategies such as Logit Masking to constrain the output search space and a Gloss-augmented architecture that explicitly leverages lexical definitions. The models are fine-tuned on the SemCor corpus and evaluated against the standard Unified Evaluation Framework benchmarks. Experimental results demonstrate that while standard Transformer fine-tuning yields performance comparable to frequency-based heuristics, the Gloss-augmented DeBERTa model achieves a significant improvement, reaching an F1 score of 74.3%. The study benchmarks these findings against current state-of-the-art systems, highlighting the superior efficacy of incorporating gloss information over simple label classification, while also discussing the limitations of independent sentence-pair modeling compared to extractive and sequential approaches.

***K*eywords** Word Sense Disambiguation, Natural Language Processing, Transformers, DeBERTa, BERT, Supervised Learning, SemCor, Gloss-based WSD

1 Introduction

Natural language is inherently ambiguous, with a significant portion of words possessing multiple distinct meanings depending on the context in which they appear. Word Sense Disambiguation (WSD) is the computational task of automatically identifying the correct meaning of a word in a specific context from a predefined sense inventory, such as WordNet. WSD remains a fundamental challenge in Natural Language Processing (NLP) and serves as a critical prerequisite for numerous downstream applications, including Machine Translation, Information Retrieval, and Question Answering, where precise semantic understanding is paramount.

The field of WSD has evolved significantly in recent decades. Early approaches relied heavily on knowledge-based methods, such as the Lesk algorithm, which utilize dictionary definitions, or supervised systems dependent on manual feature engineering. However, these methods often struggled with the data scarcity problem inherent to WSD – the limited availability of large-scale, sense-annotated corpora. The advent of deep learning, and specifically the introduction of Transformer-based architectures, marked a paradigm shift. Modern Large Language Models (LLMs) and contextualized embeddings have demonstrated a superior ability to capture syntactic and semantic nuances, establishing new state-of-the-art benchmarks across various semantic tasks.

This study focuses on the implementation, fine-tuning, and comparative analysis of encoder-only Transformer architectures for the WSD task. Specifically, the research evaluates the performance of the BERT, RoBERTa, and DeBERTa models. Beyond standard fine-tuning, this work investigates architectural modifications and inference strategies, including the implementation of Logit Masking to constrain the search space to valid candidates, and the integration of gloss information (Gloss DeBERTa) to leverage lexical definitions. These neural approaches are benchmarked against classical baselines, namely the Lesk algorithm and the Most Frequent Sense (MFS) heuristic, to quantify the performance gains offered by modern deep learning techniques.

The remainder of this paper is structured as follows. It begins with an overview of existing methodologies, alongside a review of current state-of-the-art systems. Subsequently, the methodology is detailed, providing theoretical descriptions of the selected models and the specific techniques used. This is followed by a description of the implementation details covering the dataset preparation pipeline, the fine-tuning process, and the training configuration. The experimental results are then presented alongside a discussion of the findings, analyzing the efficacy of the different architectures. Finally, the study concludes with a summary of the main contributions and potential directions for future work.

2 Related Work

2.1 Taxonomy of WSD Approaches

The methodologies used for Word Sense Disambiguation (WSD) can be broadly categorized based on the resources they utilize and the nature of their training process. Generally, these techniques fall into three main categories: knowledge-based, machine learning (supervised and unsupervised), and deep learning approaches [1]. Each paradigm possesses distinct characteristics, advantages, and limitations depending on the availability of annotated data and external knowledge resources.

2.1.1 Knowledge-Based Approaches

Knowledge-based methods are highly dependent on external lexical resources, including machine-readable dictionaries and thesauri (or lexical databases) such as WordNet. These approaches extract semantic information from word definitions and hierarchical relationships between concepts to determine the correct sense. A quintessential example of this category is the Lesk algorithm, introduced in 1986, which operates on the premise of “word overlap”. It disambiguates a target word by comparing its dictionary definition with the definitions of the surrounding context words. Although knowledge-based systems are computationally efficient and do not require extensive labeled corpora, making them suitable for low-resource scenarios, they often struggle with high polysemy and lack of coverage in lexical resources. [1]

2.1.2 Supervised Approaches

Supervised approaches treat WSD as a classification task, training a classifier on a corpus where words are manually annotated with their correct senses. These methods leverage machine learning algorithms, such as Support Vector Machines (SVM), Naive Bayes, and Decision Trees, to learn patterns from labeled examples. Supervised systems effectively capture complex linguistic relationships and typically yield high accuracy when sufficient training data is available. However, their primary drawback is the “knowledge acquisition bottleneck” – the high cost and time required to produce large-scale, sense-annotated datasets. [1]

2.1.3 Unsupervised Approaches

Unsupervised methods attempt to bypass the need for manually labeled data by utilizing clustering or topic modeling techniques. The underlying assumption is that words with similar meanings appear in similar contexts. Consequently, these algorithms group context vectors into clusters, where each cluster corresponds to a distinct sense. Although this paradigm eliminates the dependency on expensive annotated corpora, detecting the correct sense for each induced cluster remains a challenge, and these methods generally achieve lower accuracy compared to supervised counterparts. [1]

2.1.4 Deep Learning Approaches

In recent years, deep learning has revolutionized the field of NLP, offering state-of-the-art performance in WSD. Neural architectures, particularly Transformers like BERT and GPT, utilize pre-trained contextualized embeddings to capture intricate syntactic and semantic patterns from vast amounts of unlabeled text. Unlike traditional static embeddings (e.g., Word2Vec), models like BERT generate dynamic representations where the vector for a word changes depending on its context, effectively addressing polysemy. These models minimize the need for manual feature engineering and have demonstrated superior capability in handling long-range dependencies and cross-lingual transfer, although they require significant computational resources for training and inference. [1]

2.2 Selected State-of-the-Art Approaches

2.2.1 WSD as Extractive Sense Comprehension

A recent paradigm shift in Word Sense Disambiguation involves reframing the task from a standard multi-label classification problem to a span extraction task, drawing inspiration from Extractive Question Answering. Barba et al. [2] argued that traditional classification heads operating over a fixed sense vocabulary limit a model’s ability to generalize to rare or unseen senses and fail to fully exploit the semantic information explicitly provided in glosses. To address these limitations, they proposed the Extractive Sense Comprehension (ESC) framework.

In this formulation, the model receives a concatenation of the context sentence containing the target word and a sequence of all its possible definitions derived from the sense inventory. The objective is effectively transformed into identifying the text span that corresponds to the correct definition. The architecture implementing this approach, named ESCHER, utilizes a Transformer backbone (e.g., BART) to jointly encode the context and the candidate definitions.

To mitigate the inherent bias of WSD datasets towards the Most Frequent Sense (MFS), the authors introduced a regularization technique termed “Gloss Noise”. This strategy involves dynamically sampling unrelated definitions as negative candidates during training to reduce the prior probability of frequent glosses. Experimental results indicate that ESCHER establishes a new state-of-the-art on standard English WSD benchmarks, outperforming previous bi-encoder and classifier-based systems. Furthermore, the extractive nature of the model demonstrates superior data efficiency in few-shot scenarios and robust zero-shot generalization to unseen senses or different lexical resources.

2.2.2 Sequential Disambiguation with Feedback Loops

Based on the extractive formulation, Barba et al. [3] challenged the prevailing assumption in supervised WSD that words in a sentence should be disambiguated independently of each other. They argued that human understanding is a continuous process in which the meaning of a word helps clarify the meaning of subsequent words. To bridge this gap, they proposed ConSeC (Continuous Sense Comprehension), a framework that introduces a feedback loop into the inference process.

In this approach, the disambiguation targets within a text are sorted according to a specific criterion, such as increasing polysemy. The model then processes words sequentially; crucial to this method is that the prediction for a current target word is conditioned not only on its local context and candidate definitions, but also on the explicit definitions of the senses assigned to previously disambiguated words. To integrate this additional information without disrupting the natural flow of the sentence, the authors utilized the DeBERTa architecture, using its disentangled attention mechanism. Specifically, they manipulated the relative position embeddings to make the model perceive the glosses of context words as being located immediately next to the words they define, effectively injecting semantic knowledge directly into the context stream.

Empirical evaluations demonstrated that this sequential strategy significantly outperforms independent disambiguation models, achieving new state-of-the-art results on standard English benchmarks. Notably, the feedback loop proved particularly beneficial for identifying Least Frequent Senses (LFS), suggesting that explicit semantic cues from surrounding words help mitigate the model’s bias towards the most frequent sense.

3 Methodology

3.1 Task Formulation

Formally, the Word Sense Disambiguation (WSD) task can be defined as follows. Let $S = [w_1, w_2, \dots, w_n]$ be a sequence of words forming a sentence or a text document. For a specific target word $w_i \in S$, let $\Omega(w_i) = \{s_{i,1}, s_{i,2}, \dots, s_{i,k}\}$ be the predefined set of possible senses (the sense inventory) associated with w_i. The goal of a WSD system is to identify the most appropriate sense $\hat{s} \in \Omega(w_i)$ that best captures the meaning of w_i within the given context S.

This problem is typically modeled as a classification task in which a function f maps the pair (w_i, S) to a sense label:

$$\hat{s} = \underset{s \in \Omega(w_i)}{\operatorname{argmax}} P(s \mid w_i, S)$$

where $P(s \mid w_i, S)$ represents the conditional probability of sense s given the target word and its surrounding context.

3.2 Baselines

To evaluate the effectiveness of neural architectures, it is essential to benchmark them against established classical methods that do not rely on deep contextual representations. Consequently, two standard baselines are considered: a knowledge-based approach and a frequency-based heuristic.

3.2.1 Lesk Algorithm

The Lesk algorithm is a seminal knowledge-based method for WSD that relies on the hypothesis that words used together in a sentence are likely to share a common topic. Consequently, their dictionary definitions (glosses) should share overlapping vocabulary. [4]

In its simplified version, for a target word w_i and a context window C, the algorithm computes a score for each candidate sense $s \in \Omega(w_i)$ by calculating the word overlap between the gloss of s, denoted as $g(s)$, and the context C. The sense with the highest overlap score is selected:

$$\text{score}_{Lesk}(s) = |g(s) \cap C|$$

where $|\cdot|$ denotes the cardinality of the intersection set (i.e., the number of common words). This method does not require labeled training data but depends heavily on the quality and length of the dictionary definitions. [5]

3.2.2 Most Frequent Sense (MFS)

The Most Frequent Sense (MFS) heuristic serves as a powerful supervised baseline due to the highly skewed distribution of word senses in natural language (often following a Zipfian distribution). This approach assumes that for any given ambiguous word, one dominant sense accounts for the majority of its occurrences.

The MFS system simply assigns the sense that appears most frequently in a reference sense-annotated corpus (e.g., SemCor) to every instance of the target word in the test set, disregarding the specific local context:

$$\hat{s}_{MFS} = \underset{s \in \Omega(w_i)}{\operatorname{argmax}} \operatorname{count}(s, \mathcal{D}_{train})$$

Despite its simplicity, MFS is notoriously difficult to beat for many WSD systems, particularly for words with high polysemy where the dominant sense is overwhelmingly probable.

3.3 Transformer-based Models

The advent of the Transformer architecture has fundamentally altered the landscape of Natural Language Processing. For discriminative tasks such as Word Sense Disambiguation, encoder-only Transformer models have proven particularly effective due to their ability to generate deep contextualized representations. Unlike static embeddings, which assign a fixed vector to each word type, these models utilize a self-attention mechanism to compute dynamic representations based on the entire input sequence. This section details the specific architectures and modifications employed in this study.

3.3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) utilizes a multi-layer bidirectional Transformer encoder. The model is pre-trained on large corpora using two unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In the context of WSD, BERT takes a tokenized sentence $X = (x_1, \ldots, x_T)$ as input and outputs a sequence of hidden vectors $H = (h_1, \ldots, h_T)$. The hidden state h_i corresponding to a target word w_i aggregates information from the entire context, allowing a classification head to predict the correct sense label based on the contextual nuance rather than the surface form alone. [6]

3.3.2 RoBERTa

RoBERTa (Robustly Optimized BERT approach) builds upon the BERT architecture but introduces critical modifications to the pre-training procedure. Specifically, it eliminates the Next Sentence Prediction task, which was found to be redundant, and employs dynamic masking, where the masking pattern is generated anew for each input sequence during training. Furthermore, RoBERTa is trained with significantly larger batch sizes and on more extensive data. These optimizations generally result in more robust representations compared to the original BERT model, theoretically providing a better initialization point for fine-tuning on semantic tasks. [7]

3.3.3 DeBERTa

DeBERTa (Decoding-enhanced BERT with disentangled attention) improves upon BERT and RoBERTa by introducing a disentangled attention mechanism and an enhanced mask decoder. Unlike previous models where content and position information are summed into a single vector, DeBERTa encodes word content and relative position separately.

For a query token i and a key token j, the attention score $A_{i,j}$ is computed using disentangled matrices. Let H_i represent the content vector and $P_{i|j}$ represent the relative position vector. The attention weight is calculated as the sum of content-to-content, content-to-position, and position-to-content dependencies:

$$A_{i,j} = H_i^T H_j + H_i^T P_{j|i} + P_{i|j}^T H_j$$

This disentanglement allows the model to distinguish between semantic content and syntactic positioning more effectively, which is crucial for distinguishing subtle sense distinctions in WSD. [8]

3.3.4 Logit Masking DeBERTa

Standard fine-tuning of WSD models often treats the problem as a classification task over the entire sense inventory (e.g., all synsets in WordNet). However, this is computationally inefficient and linguistically inaccurate, as a target word w can only map to a small subset of valid senses $\Omega(w)$.

To address this, the Logit Masking strategy constrains the search space during both training and inference. For a given target word w, a mask vector M is constructed for the output logits layer. Let $z \in \mathbb{R}^{|V|}$ be the raw logits predicted by the model, where $|V|$ is the total size of the sense inventory. The mask is defined as:

$$M_k = \begin{cases} 0 & \text{if } s_k \in \Omega(w) \\ -\infty & \text{otherwise} \end{cases}$$

The probability distribution over senses is then computed via a masked softmax:

$$P(s_k|w) = \frac{\exp(z_k + M_k)}{\sum_{j=1}^{|V|} \exp(z_j + M_j)}$$

This forces the probability of invalid senses to zero, ensuring that the model focuses exclusively on disambiguating between the linguistically plausible meanings of the target word.

3.3.5 Gloss-augmented DeBERTa (Gloss DeBERTa)

While standard classification models rely solely on sense labels, Gloss-augmented approaches leverage the semantic definitions (glosses) provided by the knowledge base. This reformulates WSD as a sentence-pair classification task (or context-gloss pair scoring).

In this architecture, the input sequence is constructed by concatenating the context sentence S and a candidate gloss g associated with the target word, separated by a special token:

$$\text{Input} = \texttt{[CLS]}\ S\ \texttt{[SEP]}\ g\ \texttt{[SEP]}$$

The DeBERTa encoder processes this sequence, allowing the self-attention mechanism to directly model the interaction between the target word in context and the definition words. The final hidden state corresponding to the [CLS] token is fed into a binary classifier to predict a relevance score $y \in [0, 1]$, indicating whether gloss g is the correct sense for the target word in S. During inference, the model scores all candidate glosses $g \in \Omega(w)$ and selects the one maximizing the probability. [9]

4 Implementation Details

4.1 Technological Stack

The computational experiments for this study were conducted using a combination of cloud-based and on-premise high-performance computing resources. Initial prototyping and preliminary model fine-tuning were performed on Google Colab utilizing NVIDIA Tesla T4 GPUs. The primary training phases and extensive comparative experiments were executed on the ClusterFIT university computing cluster, leveraging high-end NVIDIA Tesla A100 and V100 GPUs to handle the computational load of large Transformer models.

The implementation relies on the standard Python machine learning ecosystem and Natural Language Toolkit (NLTK) library. Data Preparation, Data Analysis, and Visualization were conducted using Jupyter Notebooks, while the core training loops were encapsulated in modular Python scripts. Pre-trained model weights and tokenizers were sourced from the Hugging Face Hub.

4.2 Dataset Preparation

A robust data pipeline was developed to standardize the input format across different corpora, ensuring compatibility with the tokenization requirements of Transformer-based architectures.

4.2.1 Training Data

The training dataset was constructed from SemCor, the largest manually sense-annotated corpus available for English, accessed via the NLTK library. A custom processing script was implemented to transform the corpus from its native chunked format into a structured dataset suitable for token classification.

The pipeline iterates through the tagged sentences of the corpus, dynamically reconstructing the full raw text while simultaneously tracking the character-level indices of every token. This precise offset tracking is critical for mapping target words to their corresponding sub-word tokens after BERT-style tokenization.

For every annotated chunk, the pipeline extracts the semantic label. Since SemCor annotations often refer to specific lemmas, these were mapped to their corresponding WordNet synset names (e.g., mapping a specific lemma instance to *dog.n.01*) to create a consistent label space. Instances where the label could not be resolved to a valid WordNet synset were discarded. The resulting dataset consists of entries containing the full sentence, the target word, its character start and end positions, and the synset label. Finally, the string labels were mapped to integer IDs and the processed data was stored in Parquet format for efficient I/O operations.

4.2.2 Dataset Analysis

Following the conversion of the SemCor corpus, a statistical analysis was conducted to understand the characteristics and potential biases of the training data. The processed dataset comprises 224,515 training instances, covering approximately 33,000 unique word types and 25,805 distinct synset labels.

The distribution of Parts of Speech (POS) is heavily skewed. As illustrated in Figure 1, nouns and verbs constitute the vast majority of the annotated instances, whereas adjectives and adverbs are significantly underrepresented.

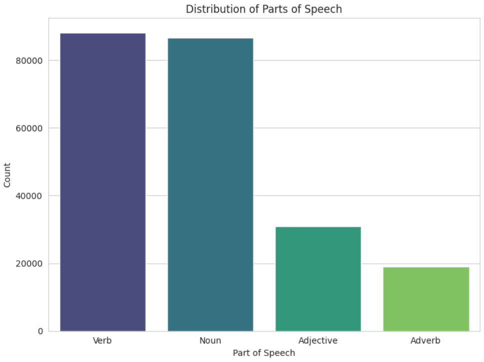

Figure 1: Distribution of Parts of Speech in the training dataset

A critical challenge identified in the dataset is severe class imbalance. As depicted in Figure 2, a small number of frequent senses, such as *be.v.01* and *person.n.01*, appear thousands of times. Conversely, the distribution exhibits a long tail, where the majority of valid senses have very few associated training examples. This sharp drop-off suggests that even when words possess multiple meanings, one sense often overwhelmingly dominates. Consequently, this imbalance is likely to bias the model toward these frequent senses, potentially hindering its ability to recognize rarer definitions in real-world scenarios.

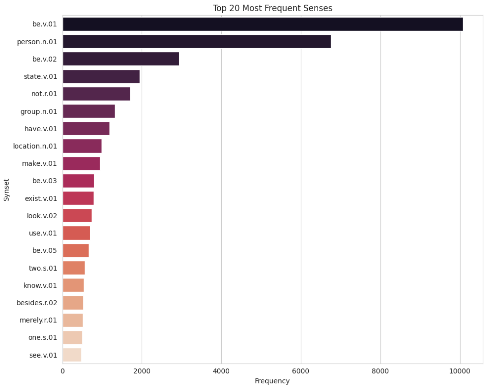

Figure 2: The 20 most frequent senses in the training dataset

Finally, the distribution of sentence lengths was examined. Figure 3 displays this distribution, which reveals that the average sentence length is approximately 30 words. This indicates that standard Transformer input sequences are sufficient to capture the full context for the majority of samples.

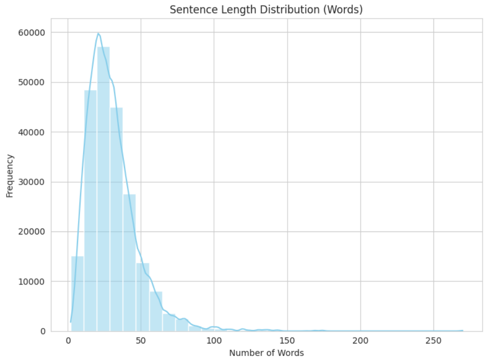

Figure 3: Distribution of sentence lengths

4.2.3 Evaluation Data

To benchmark the models against the state-of-the-art, the study utilizes the standard Unified Evaluation Framework datasets originally compiled by Raganato et al. [10]. The specific datasets included in the evaluation are **Senseval-2**, **Senseval-3**, **SemEval-2007**, **SemEval-2013**, and **SemEval-2015**, alongside the concatenated **ALL** dataset.

The test data was sourced from an external repository[1] which provides the corpora in JSONL format, converted from the original XML. A preprocessing routine was developed to harmonize these files with the training data format. A key step in this process involved label normalization: the benchmark datasets typically provide ground truth in the form of WordNet *Sense Keys* (e.g., `art%1:09:00::`). These keys were programmatically converted into Synset names (e.g., *art.n.03*) using the WordNet interface.

Furthermore, unlike the training set where each instance typically has a single label, the evaluation datasets may contain multiple valid senses for a single target word. The pipeline accommodates this by storing *gold labels* as lists. Character offsets were also recalculated from token indices to ensure consistency with the training data format. The final processed evaluation sets were saved as Parquet files.

4.3 Baseline Implementations

To provide a robust point of comparison for the neural architectures, three non-neural baseline systems were implemented using standard lexical resources.

4.3.1 Lesk Algorithm

The study utilized the implementation of the Original Lesk algorithm provided by the Natural Language Toolkit (`nltk`). This knowledge-based approach disambiguates a target word by calculating the overlap between its context (the surrounding words in the sentence) and the definitions (glosses) of its candidate synsets in WordNet. The candidate with the highest overlap score is selected as the prediction.

4.3.2 Most Frequent Sense (MFS)

The standard MFS baseline was implemented by leveraging the frequency statistics inherent in the WordNet structure. Within WordNet, the synsets associated with a specific lemma are ordered by their frequency of occurrence in the SemCor corpus. Consequently, the MFS system was implemented to simply predict the first synset in the ordered list returned by WordNet for the target lemma, disregarding the local context.

4.3.3 MFS with POS Filtering (MFS+POS)

A refined variant of the frequency baseline was implemented to incorporate syntactic constraints. Unlike the standard MFS, which might predict a noun sense for a word acting as a verb (e.g., selecting the noun “run” when the target is the verb “to run”), this approach first filters the candidate synsets based on the Part-of-Speech (POS) of the target word obtained from WordNet. After filtering, the system selects the most frequent synset among those matching the correct grammatical category.

4.4 Model Specifications

This section details the specific hyperparameters, training configurations, and architectural modifications applied to each model. All models were initialized using pre-trained weights from the Hugging Face Hub.

4.4.1 Standard Classification Models

The BERT, RoBERTa, and standard DeBERTa models were fine-tuned as token classifiers. The training was performed using the AdamW optimizer with a linear learning rate scheduler.

For **BERT**, the `bert-base-uncased` checkpoint was used. Training was conducted on Google Colab using an NVIDIA Tesla T4 GPU, completing in approximately 2 hours and 16 minutes. Similarly, **RoBERTa** utilized the `roberta-base` checkpoint on the same hardware, with a training time of 2 hours and 21 minutes.

The **DeBERTa** model employed the `microsoft/deberta-v3-base` checkpoint. Unlike the previous models, this training was executed on the university’s ClusterFIT infrastructure (utilizing Tesla A100/V100 GPUs), resulting in a reduced training time of 1 hour and 52 minutes. A warmup ratio of 0.1 was

[1] `https://github.com/akranlu/wsd-data`

introduced for DeBERTa to stabilize the early training phases.

The specific hyperparameters for these models are summarized in Table 1.

4.4.2 DeBERTa with Logit Masking

The Logit Masking experiment utilized the same base architecture and hyperparameters as the standard DeBERTa model (see Table 1). The training time on ClusterFIT was 1 hour and 47 minutes.

The masking mechanism was implemented by modifying the output logits prior to the softmax activation. For each target word, a mask vector was constructed where valid candidate senses – derived from WordNet based on the target lemma – were assigned a value of 0. All other indices in the vocabulary were assigned a large negative value (-10000) to effectively zero out their probabilities. To ensure numerical stability and correct loss calculation, the mask was explicitly set to 0 for the ground truth label, ensuring the correct sense is always a valid candidate even if it is missing from the retrieved synset list.

4.4.3 Gloss DeBERTa

The Gloss-augmented approach required a fundamentally different data preprocessing pipeline and training configuration. To facilitate the sentence-pair classification task, the original training dataset was transformed into an expanded format. For every target word instance, multiple training rows were generated corresponding to each possible WordNet synset candidate.

The input sequence was structured by enclosing the target word in the context sentence with special `[TGT]` tokens to signal the focus of disambiguation. This modified sentence was then concatenated with the candidate gloss. The resulting structure allows the model to assess the semantic compatibility between the marked word and the definition:

Context: The Fulton County Grand Jury `[TGT]` said `[TGT]` ...
Gloss: express in words
Candidate: state.v.01
Label: 1 (Positive)

Conversely, incorrect candidates (e.g., *allege.v.01*) were labeled as 0. The test data underwent an identical transformation to match the training data pattern.

Due to the increased input length resulting from the concatenation of context and glosses, the maximum sequence length was increased to 256. The learning rate was lowered to $2\text{e}{-5}$, and the number of epochs was reduced to 3. The training took approximately 8 hours and 8 minutes on ClusterFIT.

This formulation introduces a severe class imbalance, as the number of incorrect candidate glosses significantly outnumbers the correct one for any given word. To prevent the model from converging to a trivial solution (always predicting 0), a custom Trainer was implemented with a weighted Cross-Entropy Loss. The loss function penalized errors on positive examples ten times more heavily than on negative ones:

$$\mathcal{L} = \text{CrossEntropyLoss}(w = [1.0, 10.0])$$

This weighting strategy ensures that the model learns to identify the correct definition rather than simply rejecting all candidates.

5 Results and Discussion

5.1 Evaluation Metrics

The primary metric employed for evaluation is the F1 Score. In the context of Word Sense Disambiguation, where every token in the test set is assigned exactly one label and the task is formulated as multi-class classification, the Micro-averaged F1 Score is mathematically equivalent to Accuracy. Consequently, the results are reported as F1 Score (Micro), representing the percentage of correctly disambiguated target words.

5.2 Comparative Analysis

The experimental results on the Unified Evaluation Framework datasets are summarized in Table 2. The table compares the performance of the baseline algorithms against the standard transformer models and the specialized architectural modifications.

5.2.1 Baselines Performance

The knowledge-based Lesk algorithm yielded the lowest performance across all datasets (33.6% on ALL), confirming the limitations of relying solely on definition overlaps without learning statistical patterns. The Most Frequent Sense (MFS) heuristic provided a strong baseline (49.1%), demonstrating the highly skewed nature of word senses. notably, the addition of Part-of-Speech filtering (MFS+POS) resulted in a substantial improvement (+9.3% on ALL), achieving 58.4%. This result highlights that simply constraining the search space to the correct grammatical category solves a significant portion of ambiguity.

5.2.2 Standard Transformers vs. Baselines

A key observation from the experiments is that standard fine-tuning of encoder-only models (BERT, RoBERTa, DeBERTa) yields results that are statistically comparable to each other, hovering around 59% on the combined dataset. Surprisingly, these sophisticated neural models only marginally outperform the simple MFS+POS heuristic. For instance, on the SemEval-2015 dataset, the standard DeBERTa model (59.6%) actually underperformed compared to MFS+POS (60.7%).

This stagnation can be attributed to the vast output space. With over 25,000 potential synsets in the classification

Table 1: Hyperparameters and training details for the encoder-only models.

Model	Base Model	LR	Batch	Seq Len	Epochs	Time
BERT	bert-base-uncased	5e−5	16	128	5	2h 16m
RoBERTa	roberta-base	5e−5	16	128	5	2h 21m
DeBERTa	deberta-v3-base	5e−5	16	128	5	1h 52m

Table 2: F1 Score performance on the standard English WSD benchmarks

Model	Senseval-2	Senseval-3	SemEval-07	SemEval-13	SemEval-15	ALL
Baselines						
Lesk	35.5	31.0	21.1	36.3	35.4	33.6
MFS	49.4	47.1	39.6	53.4	49.4	49.1
MFS + POS	60.9	59.0	51.0	54.9	60.7	58.4
Standard Transformers						
BERT (base)	61.1	62.2	59.8	52.4	60.6	59.2
RoBERTa (base)	60.8	61.6	60.0	52.1	60.6	58.9
DeBERTa (base)	60.7	61.7	60.0	51.8	59.6	58.8
Specialized Architectures						
DeBERTa + Logit Masking	64.9	67.3	62.6	62.2	66.1	65.0
Gloss DeBERTa	**74.0**	**71.0**	**68.6**	**77.5**	**78.1**	**74.3**

head, the models struggle to discriminate effectively, often distributing probability mass across linguistically invalid senses for a given target word.

5.2.3 Impact of Logit Masking

The application of Logit Masking to the DeBERTa model addressed the issue of the large search space. By constraining the model to choose only from senses compatible with the target lemma, performance increased significantly, from 58.8% to 65.0% on the ALL dataset. This +6.2% gain confirms that the model's internal representations are often correct, but the standard classification head is prone to predicting unrelated senses in the absence of constraints.

To further investigate the behavior of the masked model, an error analysis was conducted on instances where the standard DeBERTa model predicted the correct sense while the Logit Masking model failed. This analysis revealed that in approximately 98% of these divergent cases, the correct sense was indeed present in the mask (i.e., it was a valid candidate). This counter-intuitive finding suggests a potential downside to the masking strategy: by training on a significantly reduced search space for each word, the model may become "lazy" or less discriminative. Because it is never forced to distinguish the correct sense from the full spectrum of 25,000 global classes during training, its decision boundaries within the small subset of valid senses may be less robust than those of the unmasked model in specific ambiguous contexts.

5.2.4 Superiority of Gloss-based Approaches

The Gloss DeBERTa architecture achieved state-of-the-art performance in these experiments, reaching an F1 score of 74.3% on the concatenated dataset. This represents a dramatic improvement of nearly 10% over the masked classification approach and 15% over standard fine-tuning. This performance leap validates the hypothesis that WSD is better modeled as a sentence-pair classification task (Context-Gloss matching) rather than a simple token classification task. By explicitly encoding definitions, the model can generalize better to rare senses (the "long tail" of the distribution) that appear infrequently in the training data, as it learns to match semantic similarity rather than memorizing class labels.

5.3 Comparison with State-of-the-Art

To contextualize the performance of the best-performing model (Gloss DeBERTa) developed in this study, it is benchmarked against recent State-of-The-Art (SoTA) systems that also utilize the SemCor corpus for training. Specifically, the analysis focuses on **ESCHER** [2], which treats WSD as an extractive span prediction task, and **ConSeC** [3], which extends the extractive approach with a sequential feedback loop mechanism.

Table 3 presents the comparative results on the ALL benchmark dataset.

Table 3: Comparison of the Gloss DeBERTa model with SoTA systems

Model	F1 Score (ALL)
Gloss DeBERTa	74.3
ESCHER	80.7
ConSeC	82.0

While the Gloss DeBERTa implementation significantly outperforms standard classification baselines and Logit Masking approaches, it lags behind the specialized SoTA architectures. ESCHER achieves an F1 score of 80.7%, surpassing the Gloss DeBERTa model presented here by 6.4 percentage points. This gap highlights the efficacy of the extractive formulation over the sentence-pair binary classification approach used in this work. By predicting the correct definition directly from a concatenated sequence of all candidates (as in ESCHER), the model can likely compare candidates more effectively in a single pass rather than scoring them independently.

Furthermore, ConSeC achieves the highest performance with 82.0%, demonstrating the value of sequential context integration. The current implementation processes each target word independently. The superior results of ConSeC suggest that incorporating the resolved meanings of surrounding words into the context of the current target word is a critical factor for achieving human-level disambiguation performance.

6 Conclusion

This study successfully achieved its primary objectives: to implement, fine-tune, and evaluate Transformer-based architectures for the task of Word Sense Disambiguation. The research followed a structured progression from establishing classical baselines to fine-tuning neural models, culminating in the development of specialized architectures that leverage gloss information.

The experimental results definitively demonstrate that while standard encoder-only Transformers (BERT, RoBERTa, DeBERTa) provide a solid foundation for semantic analysis, simply treating WSD as a standard classification task over a large vocabulary is suboptimal. The findings highlight that significant performance gains are achieved not merely by changing the backbone model, but by reformulating the task itself. The implementation of Logit Masking provided a substantial improvement by incorporating linguistic constraints, while the Gloss DeBERTa approach yielded the best performance (74.3% F1), confirming that explicitly modeling the relationship between context and definition is key to resolving semantic ambiguity. Although the implemented solution did not surpass the absolute top-performing systems like ESCHER or ConSeC, it demonstrated a competitive capacity to generalize to rare senses, significantly outperforming strong frequent-sense baseline.

6.1 Future Work

Several avenues remain open for future research to further bridge the gap to human-level performance. First, the experiments in this study were constrained to ‘base’-sized models due to computational limits. Scaling up to ‘large’ or ‘xl’ variants of DeBERTa would likely yield immediate improvements in representational power and accuracy.

Second, the dependence on the SemCor corpus, which is relatively small and unbalanced, remains a bottleneck. Future work could investigate the inclusion of “silver” data (automatically annotated corpora) or the utilization of data augmentation techniques to address the scarcity of examples for rare senses.

Finally, the architectural comparison suggests that the next logical step is to move beyond independent sentence-pair classification. Adopting extractive frameworks (span prediction) and integrating sequential decoding mechanisms – where the disambiguation of one word informs the next – appears to be the most promising direction for capturing the full coherent meaning of a text.

References

[1] Robbel Habtamu and Beakal Gizachew. State-of-the-art approaches to word sense disambiguation: A multilingual investigation. In *Pan-African Conference on Artificial Intelligence*, pages 176–202. Springer Nature Switzerland, 2024.

[2] Edoardo Barba, Tommaso Pasini, and Roberto Navigli. ESC: Redesigning WSD with extractive sense comprehension. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online, June 2021. Association for Computational Linguistics.

[3] Edoardo Barba, Luigi Procopio, and Roberto Navigli. ConSeC: Word sense disambiguation as continuous sense comprehension. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[4] Michael 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*, SIGDOC ’86, page 24–26, New York, NY, USA, 1986. Association for Computing Machinery.

[5] Adam Kilgarriff and Joseph Rosenzweig. English senseval: Report and results. In M. Gavrilidou, G. Carayannis, S. Markantonatou, S. Piperidis, and G. Stainhauer, editors, *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC’00)*, Athens, Greece, May 2000. European Language Resources Association (ELRA).

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[8] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654, 2020.

[9] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. Glossbert: BERT for word sense disambiguation with gloss knowledge. *CoRR*, abs/1908.07245, 2019.

[10] Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. Word sense disambiguation: A unified evaluation framework and empirical comparison. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain, April 2017. Association for Computational Linguistics.