Bi-Encoder-Siamese-Networks-For-Word-Sense-Disambiguation

Team: Spacey Force

Aditi Gupta (2024701024)

Amit Shukla (2024701035)

Nikhil Sivakumar (2023114018)

1 Introduction

Word Sense Disambiguation (WSD) is a long-standing and crucial challenge in Natural Language Processing (NLP) that focuses on figuring out which meaning of a word is intended in a particular sentence or context, especially when the word has multiple meanings. Getting this right is crucial for many applications like machine translation, search engines, text summarization, and understanding the meaning of texts.

While modern models like BERT are great at capturing general meaning, they often miss the fine details needed to tell apart subtle word senses. To address this, many recent methods fine-tune these models on specialized WSD datasets, usually using encoder-only architectures.

In this work, we introduce a new approach: a **Siamese Bi-Encoder Network**. Unlike cross-encoders that process both inputs together (and can be slow), our bi-encoder setup processes the sentence context and the sense definition (gloss) separately, making it much faster and more scalable. Siamese architectures are particularly good at tasks that require comparing or matching two inputs because they use identical subnetworks with shared weights — this leads to consistent, efficient learning and fewer parameters to train. This makes them especially useful for WSD, including in low-resource or contrastive learning setups.

We train our model in two main stages:

- **Pretraining:** We first teach the model through several supervised tasks that help it learn the relationships between sentence contexts and sense definitions, such as:
  - Mapping the sentence context to the correct sense (gloss),
  - Aligning the context with broader categories (hypernyms),
  - Discriminating between correct and incorrect context-gloss pairs using triplets.
- **Fine-tuning:** Next, we further train the bi-encoder using well-known datasets like SemCor and WiC, focusing on sentence pairs or triplet setups to fine-tune its ability to judge semantic similarity at the sentence level.

Overall, our approach helps the encoders build a deep understanding of word meanings by combining knowledge from lexical resources during pretraining and learning fine-grained contextual similarities during fine-tuning. Thanks to the bi-encoder design, the model can be efficiently applied in real-world systems that need accurate, sense-aware semantic representations.

2 Related Work

2.1 Word Sense Disambiguation (WSD)

Word Sense Disambiguation has a rich history in natural language processing, with early approaches relying heavily on supervised learning over sense-annotated corpora (e.g., SemCor). Knowledge-based systems such as Lesk's algorithm and its variants utilize lexical resources like WordNet to perform disambiguation based on sense definitions and overlaps with context. However, these methods are constrained by the limited coverage and quality of lexical definitions.

Supervised neural approaches have advanced the field by leveraging sense-annotated datasets to train models that learn contextual clues for sense discrimination. The advent of contextual language models (e.g., BERT, RoBERTa) further improved the effectiveness of WSD systems. Notable efforts include fine-tuning BERT on SemCor and leveraging gloss embeddings derived from WordNet to resolve ambiguous terms. However, most of these models either use encoder-only architectures or joint (cross-encoder) formulations, which tend to be computationally expensive at inference time.

2.2 Bi-Encoder Architectures

Bi-Encoders are neural architectures that encode two inputs independently before comparing them via a similarity function. Originally popularized in retrieval tasks and question-answering systems (e.g., Dual-Encoders for passage retrieval), Bi-Encoders provide a natural fit for WSD, where the goal is to match a sentence (context) with a relevant sense (gloss).

Unlike cross-encoders that concatenate inputs and jointly compute attention across them, Bi-Encoders allow for independent computation of context and gloss embeddings. This not only improves scalability (as gloss embeddings can be precomputed and cached) but also supports modularity and transferability. However, Bi-Encoders tend to underperform cross-encoders in terms of raw accuracy, particularly without careful training on high-quality semantic objectives.

2.3 Multi-Stage Training Objectives

Recent work has explored various multi-stage and curriculum-based training strategies to enhance context-sensitive lexical understanding. Approaches such as contrastive learning using triplets [2], supervised alignment of lexical entries with sentences [1], and the use of datasets like WiC [3] have demonstrated the effectiveness of targeted supervision in capturing subtle meaning distinctions.

Building on these insights, our approach adopts a multi-stage training pipeline that progressively enhances the model's semantic sensitivity. In the pretraining stage, we integrate diverse lexical-semantic supervision signals, including context-gloss alignment,

hypernym-based hierarchical information, and sense-level contrastive triplets. These signals guide the Bi-Encoder in learning a more structured and interpretable embedding space. In the fine-tuning phase, we further refine this space using contextually grounded sentence pair similarity tasks, enabling the model to handle both discrete sense distinctions and fluid contextual shifts in meaning. This staged setup not only improves generalization across lexical tasks but also strengthens performance on few-shot and zero-shot evaluations involving sense disambiguation and contextual inference.

Dataset and Preprocessing

Consistent with most recent work, we use **SemCor 3.0**, WiC as our primary training corpus. SemCor consists of 226k sense tags and forms the largest manually annotated training corpus available. We prepare three different datasets using the same weak su-

Context Sentence	Synset Gloss Definition	Label
Styka blew his " nose " again.	nose : the organ of smell and entrance to the respiratory tract; the prominent part of the face of man or other mammals	1
Styka blew his " nose " again.	nose : search or inquire in a meddlesome way	0
Styka blew his " nose " again.	nose : defeat by a narrow margin	0

Context Sentence	Hypernym Gloss Definition	Label
Styka blew his " nose " again.	spout : an opening that allows the passage of liquids or grain	1
Styka blew his " nose " again.	search : search or seek	0
Styka blew his " nose " again.	get the better of : win a victory over	0

Context Sentence (Anchor)	Gloss Definition (Positive)	Gloss Definition (Negative)
Styka blew his " nose " again.	nose : the organ of smell and entrance to the respiratory tract; the prominent part of the face of man or other mammals	nose : search or inquire in a meddlesome way
Styka blew his " nose " again.	nose : the organ of smell and entrance to the respiratory tract; the prominent part of the face of man or other mammals	nose : defeat by a narrow margin

Figure 1: Context-Gloss Pairs, Context-Hypernym Pairs, and Context-Positive Gloss-Negative Gloss Triplets with Weak Supervision

pervision for our training experiments. Figure 1 contains example rows corresponding to each of the datasets described in the following sections.

Context-Gloss Pairs

We prepared our context-gloss pairwise dataset. For each target word within a context sentence, we retrieve all senses and their corresponding gloss. Thus, for a target word with n different senses, we have n pairwise examples for training. The pair corresponding to the correct sense (gold synsets) are positive examples (label 1), while all other pairs are negative examples (label 0) in our dataset. We generated around **79k** samples from raw SemCor-3.0 for this dataset.

Context-Hypernym Gloss Pairs

For each gold (correct) synset, we take the set of immediate hypernyms. The hypernym gloss is augmented with weak signals similar to the target synset gloss. These constitute the positive pairs in our context-hypernym gloss pairs dataset. Similarly, hypernyms of incorrect synsets for a target word in a context are negative samples in this data. We generated around **59k** samples from raw SemCor-3.0 for this dataset.

Context-Positive Gloss-Negative Gloss Triplets

We use another formulation of our datasets for optimizing on the triplet loss objective. Here, the context sentence is used as an anchor, the gloss corresponding to the correct (gold) synset for the target word is the positive example, whereas glosses corresponding to each synset for the target lemma that are not the positive synset are used as negative examples. Thus, for a target word with n different senses, we get $n-1$ training triplets in the anchor-positive-negative format, where the anchor and positive are the same across each triplet, while the negative varies. We use all the incorrect senses for each anchor-positive pair while preparing this dataset. We generated around **69k** context-pos-gloss-neg-gloss samples from raw SemCor-3.0 for this dataset.

WiC (Words In Context)

The **Word-in-Context** (WiC) dataset is a benchmark for evaluating context-sensitive word representations. Introduced by Pilehvar and Camacho-Collados (2019), it focuses on determining whether a target word retains the same meaning across two different sentences. Each example includes a target word and two sentences, and the task is a binary classification: does the word have the same sense in both contexts? WiC is built from high-quality lexical resources like WordNet and Wiktionary, and it challenges models to go beyond surface-level similarity to capture fine-grained semantic distinctions. For our purpose, we use **4,000** sentence pairs from the dataset.

SemCor-3.0 (Triplets and Pairs)

Semcor dataset has been a crucial part of our experiments as we also used to produce new derivatives as well. In this part of the study, we directly used the Semcor in its raw format, taking around **42k** context triplets and **10k** sentence pairs for fine-tuning.

Oversampling

Our final pairwise datasets are skewed towards the negative class, as we take all the negative glosses for a target lemma, which form the negative examples in our dataset. Each context sentence, however, has only one positive example for a target word. We augmented the positive class by two methods. First we add the same context sentence multiple time (generally 2-3). At the same time, to introduce variation, we use back-translation method using **paraphrase-multilingual-MiniLM-L12-v2** model from sentence transformers library. For each context, we generate 3-4 such translations. This ensures an overall balance between negative and positive samples, at the same time, enriching our data quality.

To our context-gloss datasets, the gloss definitions of different synsets with the same lemma are added to maximize the distance between gloss definitions themselves.

3 Methodology

3.1 Bi-Encoder Siamese Networks

In our approach to solving the Word Sense Disambiguation (WSD) problem, we use Siamese networks with encoder models such as **TinyBERT** and **DistilBERT**. We chose these models because they are well-suited for downstream tasks and retain many features from the original large BERT model, while being smaller and faster.

Model Specifications and Comparison

Feature	**BERT (Base)**	**DistilBERT**	**TinyBERT**
Layers (Transformer)	12	6	4 / 6 (varies)
Hidden Size	768	768	312 / 768
Attention Heads	12	12	12
Parameters	∼110M	∼66M	∼14M (TinyBERT-4)
Training Objective	MLM + NSP	MLM + Knowledge Distillation	MLM + Layer-wise Distillation
Speed	Baseline	∼60% faster	∼7.5× smaller (TinyBERT-4)
Accuracy Trade-off	High accuracy	∼97% of BERT-base	∼96–98% of BERT-base
Distillation Type	N/A	Task-agnostic	Task-specific & general

Table 1: Comparison of BERT, DistilBERT, and TinyBERT

These specifications make it possible for us to efficiently train and fine-tune the models for WSD even in resource-constrained environments.

3.2 Methodology Overview

We developed two variations of **SiamBERT** (our model name), one using a contrastive loss function and the other using a triplet loss function. These loss functions are known to perform exceptionally well in disambiguation tasks, similar to their success in face verification models.

Our Siamese network is also flexible, allowing different encoder models to be used. The encoders, combined with feed-forward layers, are trained on various datasets specifically for the WSD task. As outlined below, the model is trained in two key stages: **Pre-training** and **Fine-tuning**.

Pre-training of SiamBERT

In the pre-training step, depending on the loss function used, we train the model on three datasets derived from SemCor 3.0.

- For the **contrastive loss** setup, we train both Tiny SiamBERT and Distil SiamBERT on **Context-Gloss** and **Context-Hypernymy** datasets. This helps the model

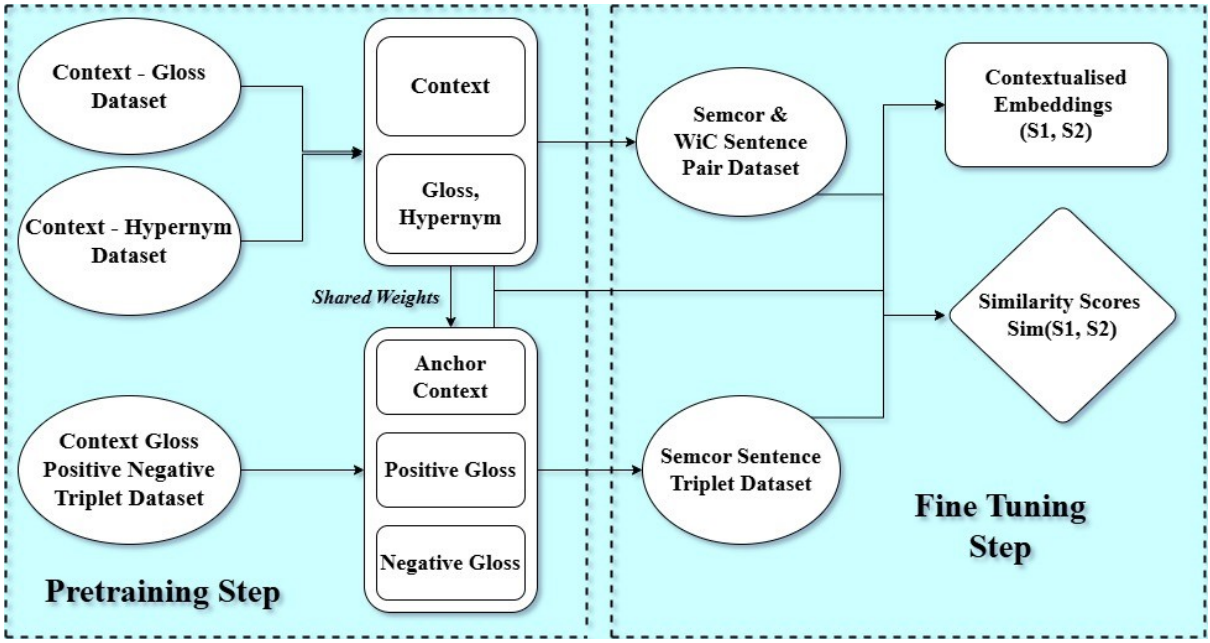

Figure 2: Pipeline Of SiamBERT Training and Fine-Tuning

learn the meanings and categories of ambiguous words across a wide range of sentences. We Maximize similarity between the sentence and its correct gloss/ hypernym (particularly useful for low-resource disambiguation scenarios), while minimizing similarity with incorrect glosses. The model outputs a similarity score from **+1** to **-1**. We have assigned **0.0** as our threshold for assigning labels in this task. The same thing has been done in the fine-tuning step too.

- For the **triplet loss** setup, we train both models on the **Context-Positive-Gloss-Negative-Gloss** dataset. Here, the model learns not only how ambiguous words are used in context, but also how to distinguish between different meanings of the same word. This ensures that the embedding of the context(C) is closer to G+(Correct gloss)than to G(Incorrect gloss).

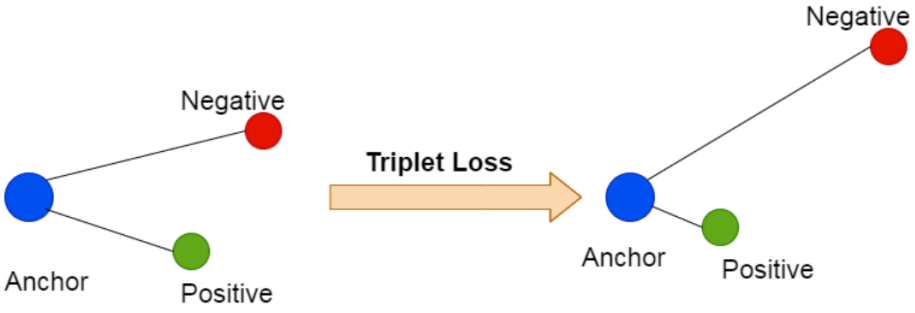

Figure 3: The triplet loss objective aims to minimize distance between the anchor and positive while maximizing the distance between anchor and negative

- Triplet Loss:

$$L = \max\left(0, sim(C, G^-) - sim(C, G^+) + \delta\right)$$

where δ is the margin hyperparameter. This fine-grained contrastive learning helps the model define precise semantic bound- aries between word senses.

- The parameters setting for pre-training step are as follows. Both models were trained for around **4-5** epochs, with optimizer set to **AdamW**, batch size of **32** and a learning rate of **2e-5** for contrastive loss and **3e-5** for triplet loss functions. The margin value for triplet loss was set to **1.0**.

Fine-tuning of SiamBERT

In the fine-tuning step, we further adapt the models to focus on the contextual meaning of ambiguous words, beyond just learning their definitions. This refines the model's ability to understand how words are used in varying contexts.

- For the **contrastive loss** model, we use a combined dataset of **WiC (Words in Context)** and SemCor 3.0, formatted as:

  Sentence 1, Sentence 2, Ambiguous Word, Label

  Here, the *Label* (0 or 1) indicates whether the ambiguous word has the same meaning in both sentences. We Predict whether the ambiguous word has the same meaning in both contexts.

  $$Sim(S_1, S_2) = \cos(E_1, E_2)$$

  where E_1 and E_2 are embeddings of S_1 and S_2, respectively.

- For the **triplet loss** model, we fine-tune using **SemCor Triplets**, which include:

  Anchor Sentence(S), Positive Sentence(S^+), Negative Sentence(S^-)

  All three sentences contain the ambiguous target word, allowing the model to learn how to differentiate between correct and incorrect senses. We Embed the anchor sentence closer to S^+ than to S^-. For getting the label, we take the **Euclidean distances** between anchor and positive sentence (**d1**) and between anchor and negative sentence (**d2**). Depending upon which is smaller, we assign the label to the context sentence.

- The parameters setting for pre-training step are as follows. Both models were trained for around **15** epochs, with optimizer set to **AdamW**, batch size of **32** and a learning rate of **2e-5** for contrastive loss and **3e-5** for triplet loss functions. The margin value for triplet loss was set to **5.0**

Overall, our approach enables the encoders to build strong semantic representations by combining lexical knowledge from pretraining with fine-grained contextual learning during fine-tuning, ensuring robust and efficient performance on WSD tasks.

3.3 Inference

At inference time, given a sentence and a set of candidate glosses (e.g., all senses of the target word), the model encodes the context and each gloss independently. It computes similarity scores between the context vector and each gloss vector and selects the gloss with the highest score as the predicted sense.

4 Experiments and Results

To evaluate the effectiveness of our SiamBERT models, we compare their performance against the baseline GlossBERT model fine-tuned on the SemCor 3.0 corpus. The GlossBERT model used for comparison is publicly available at: huggingface.co/kanishka/GlossBERT

We assess model performance on three benchmark datasets:

- **SemEval 2013**
- **SemEval 2015**
- **RAW-C (Related Words in Context)**

Each dataset tests the model's ability to disambiguate word senses based on contextual information. Accuracy is used as the evaluation metric. For all triplet based models, we calculate distance between embeddings of passed two sentences. We have taken threshold as **0.01** for assignment of labels.

SemEval 2015 Results

Model	**Accuracy**
GlossBERT	57.9439
Tiny-SiamBERT (contrastive)	**63.3602**
Distil-SiamBERT (contrastive)	**60.7748**
Tiny-SiamBERT (triplet)	**68.2242**
Distil-SiamBERT (triplet)	**74.7663**

Table 2: Performance on SemEval 2015 dataset

All SiamBERT variants outperform GlossBERT, with the triplet-loss Distil-SiamBERT achieving the highest accuracy of **74.77%**, which is approximately **17%** higher than the GlossBERT baseline.

SemEval 2013 Results

Model	**Accuracy**
GlossBERT	55.3333
Tiny-SiamBERT (contrastive)	**74.6442**
Distil-SiamBERT (contrastive)	**75.4150**
Tiny-SiamBERT (triplet)	**74.6667**
Distil-SiamBERT (triplet)	**77.3333**

Table 3: Performance on SemEval 2013 dataset

The SiamBERT models achieve a remarkable improvement over GlossBERT on SemEval 2013. Distil-SiamBERT (triplet) achieves the highest score of **77.33%**, outperforming GlossBERT by over **22** percentage points.

RAW-C (Related Words in Context) Results

Model	**Accuracy**
GlossBERT	61.9047
Tiny-SiamBERT (contrastive)	**73.9540**
Distil-SiamBERT (contrastive)	**73.9540**
Tiny-SiamBERT (triplet)	**62.7976**
Distil-SiamBERT (triplet)	**69.3452**

Table 4: Performance on RAW-C dataset

On the RAW-C dataset, both contrastive SiamBERT models significantly outperform GlossBERT. Interestingly, the contrastive variants perform better than the triplet variants on this dataset, suggesting that semantic relatedness (rather than sense discrimination) plays a larger role here.

Observations and Insights

- SiamBERT consistently outperforms GlossBERT across all datasets.
- Distil-SiamBERT with triplet loss achieves the highest overall performance, particularly excelling on sense-discriminative tasks (SemEval datasets).
- WordNet often assigns multiple distinct sense IDs to a single word, even when the meanings are highly similar and interchangeable. This poses a significant challenge in extracting data for ambiguous words, as closely related senses may be treated as entirely separate entries.
- The SiamBERT architecture offers flexibility by allowing the integration of various encoder models, depending on the specific use case.
- The improvements validate the benefits of the SiamBERT architecture and the utility of context-gloss and context-hypernym contrastive/triplet training.
- Tiny-SiamBERT also shows comparable performance with in some cases even beating results of Distil-SiamBERT.
- GlossBERT has been specifically trained on gloss information and thus often fails in cases where the word sense depends on context not just on its face value which well explains its incompetency in WSD task.

All relevant code files, helper functions/utilities, and datasets used for training and evaluation are provided in this repository: **Github**. All weight files can be found **here.**

In this section, we describe the experimental setup used to evaluate the effectiveness of our multi-stage Bi-Encoder training pipeline for word sense disambiguation and semantic similarity. We report implementation details, dataset statistics, training hyperparameters, and performance results on benchmark datasets.

5 Qualitative Examples

Sentence 1	**Sentence 2**	**Lemma**	**True Label (Dist.)**	**Predicted Label**
The lawyer presented his case in court.	The detective reviewed the case again to find new clues.	case	1 (0.0187)	1
He closed the deal with a handshake.	She found a great deal on shoes online.	deal	1 (0.0164)	1
He went to the bank to deposit his paycheck.	She visited the bank with her friend yesterday to withdraw.	bank	1 (0.0126)	1
The chef prepared a delicious dish.	The satellite captured a dish image from space.	dish	0 (-0.9812)	0
The coach gave a motivational speech before the game.	The coach broke down on the highway.	coach	0 (-0.9834)	0
She dropped her ring on the floor.	The ring of the phone startled everyone.	ring	0 (-0.9657)	0

6 Conclusion and Future Work

In this work, we presented a comprehensive, multi-stage Bi-Encoder training pipeline for word sense disambiguation and contextual semantic similarity. Our approach systematically integrates lexical supervision from glosses and hypernyms during pretraining and enhances contextual discrimination through sentence-level contrastive fine-tuning. The resulting Bi-Encoder model not only outperforms traditional sense-aware baselines like GlossBERT but also offers significant inference-time advantages due to independent encoding.

Through extensive experimentation and analysis, we demonstrated that:

- Gloss and hypernym supervision enables robust sense generalization.
- Sentence triplet fine-tuning is crucial for context-based sense discrimination.
- Our model maintains high semantic fidelity while scaling efficiently for real-time tasks.

6.1 Future Work

While our model shows strong performance, there are several exciting directions we can explore to enhance its capabilities and impact. Below are three major areas for future work:

1. **Cross-Lingual Word Sense Disambiguation (WSD):**
   Currently, our model is designed for English only. A valuable next step is to expand it to handle multiple languages. By leveraging multilingual encoders and aligning word senses across different languages, we can build a more universal model capable of performing WSD in a cross-lingual setting. This could benefit tasks like machine translation and cross-lingual search.

2. **Richer Glosses with External Knowledge:**
   WordNet glosses are concise and sometimes lack detailed information. To provide the model with more comprehensive semantic context, future work could explore enhancing glosses using external resources like Wikipedia, Wiktionary, or BabelNet. These enriched glosses would help the model better understand subtle word meanings and improve disambiguation accuracy.

3. **Application to Downstream NLP Tasks:**
   Our current work focuses on WSD, but the trained Bi-Encoder can be applied to a range of real-world NLP tasks. Integrating sense-aware representations into applications such as question answering, natural language inference, and information retrieval could further demonstrate the usefulness of our approach beyond word sense disambiguation.

References

[1] Liyuan Huang, Chi Wang, Kevin Wang, and Graham Neubig. Glossbert: Bert for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[2] Fangyu Liu, Yuxiang Lin, Yuxuan Liu, Yixin Shen, and Dragomir Radev. Learning semantic representations for novel words: Leveraging definitions in neural embedding models. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2021.

[3] Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.