Word Sense Disambiguation

1 Problem Setup

This study addresses **Word Sense Disambiguation (WSD)** by comparing two **baseline methods**—the **Most Frequent Sense (MFS)** approach and **Lesk's Algorithm**[3]—on the **SemEval 2013 Task #12 dataset**[4]. The MFS baseline, which selects WordNet's **most common sense** for each word, generally **outperforms** Lesk's Algorithm, which relies on **lexical overlap** between sense definitions and context. To enhance Lesk's performance, two additional approaches are proposed: a pretrained **BERT model** [2] with **cosine similarity** to capture richer contextual information, and a probabilistic **multinomial Naive Bayes model** with **TF-IDF features** for improved disambiguation. This report presents a comparative analysis of these methods, evaluating accuracy, strengths, and failure analysis.

2 Methodology

► ***Data Preprocessing.*** The preprocessing function **standardizes text** by **tokenizing** sentences with **SpaCy** [1] and an **NER model**, then selectively **normalizing tokens** to retain key context. **Proper nouns** and **abbreviations**, identified by the NER model or uppercase format, are preserved, while other tokens are **lemmatized** and **lowercased**. The tokens are then joined into a single, standardized string that balances contextual integrity with simplicity.
► ***Baseline Methods.*** The Most Frequent Sense (MFS) serves as a baseline through this project, leveraging the sense ordering in WordNet, where the most frequent sense of a word is listed as the first (#1) in the synset. MFS assumes that the first-listed sense is the most likely usage, providing a straightforward and effective disambiguation method.
► ***Proposed Method: BERT-based Lesk's Algorithm with Cosine Similarity.*** This method operates in two phases: **fine-tuning** and **evaluation**. During fine-tuning on the **development set**, BERT learns to distinguish word senses through exposure to **positive** (correct) and **negative** (incorrect) context-sense pairs. For each epoch, BERT processes batches, computing **cosine similarity scores** between context and **sense definition embeddings**. These scores inform the **loss**, optimized via **backpropagation**, allowing BERT to identify contextual cues associated with correct senses. In the evaluation phase on a **test set**, **cosine similarity replaces traditional overlap to compare embeddings,** with the sense showing the highest similarity selected as the predicted sense.
► ***Proposed Method: Multinomial Naive Bayes-based Lesk's Algorithm with TF-IDF.*** This method uses **Multinomial Naive Bayes (NB)** with **TF-IDF features** to determine the most likely sense of a word based on context. It involves two steps: **feature extraction** and **probabilistic scoring**. First, the context surrounding each target word is combined with each **candidate sense's definition** and transformed into a **TF-IDF vector**, emphasizing unique and significant terms. These vectors are then input to a Multinomial NB **classifier**, which learns **conditional probabilities** for each sense given the TF-IDF features, with **smoothing** to handle rare terms. During evaluation, **probabilistic scoring replaces traditional overlap**, assigning each context to the sense with the highest probability based on its TF-IDF representation, enabling effective disambiguation without extensive fine-tuning.

3 Experiments

► ***Experimental Setup.*** The dataset used in this project is divided into a development set of **1,450** instances and a test set of **194** instances. For the BERT-based Contextual Similarity method, training was conducted with a batch size of **8**, using a fine-tuning epoch count of **10** and a learning rate of 2×10^{-5}. For the Multinomial Naive Bayes-based method, TF-IDF vectorizer is with unigram features and the smoothing factor is **0.01**.
► ***Experimental Results.*** The results of main experiments in Tab. 1a reveal that **MFS baseline outperforms the traditional Lesk's Algorithm in terms of accuracy.** Specifically, the MFS achieves an accuracy of **67.53%**, whereas Lesk's Algorithm attains only **41.24%**. In addition to the baseline comparison, both of the proposed methods show substantial improvements over the traditional Lesk's Algorithm. The BERT-based Lesk's Algorithm improves upon Lesk, achieving an accuracy of **63.92%**, though it falls slightly below the MFS baseline. The NB-based Lesk's Algorithm also demonstrates the highest accuracy among all methods tested, reaching **73.20%**. Successful sample cases illustrating these improvements are presented in Tab.2.

Method	Accuracy (%)
MFS (baseline)	67.53
Lesk's Algo	41.24
BERT-based Lesk's Algo	65.98
NB-based Lesk's Algo	**72.30**

(a) Accuracy of each method

Tuning epoch	Accuracy (%)
5	64.95
10	**65.98**
15	58.76
20	61.34

(b) Accuracy for different epochs

alpha	Accuracy (%)
0.001	72.16
0.01	**73.20**
1.0	72.68
5.0	72.68

(c) Accuracy for different alpha

Table 1: Main experiment results and hyper-parameter tuning.

Type	BERT-based Lesk's Algorithm	NB-based Lesk's Algorithm
Success	**Context**: ... the push by small island_nation has " put political **pressure** on the entire political process." **Predicted**: pressure.n.02 **Actual**: pressure.n.02	**Context**: ... provides a range of options for the key question, including how developed countries would cut their carbon_output. **Predicted**: scope.n.01 **Actual**: scope.n.01
Failure	**Context**: John Coequyt, senior **washington** representative for the sierra_club, says ... **Predicted**: washington.n.01 **Actual**: capital.n.06	**Context**: the **united_states** will not make a deal without major developing countries ... **Predicted**: action.n.01 **Actual**: action.n.09

Table 2: Examples of Successful and Failure Cases for BERT-based and NB-based Lesk's Algorithm.

The BERT-based Lesk's Algorithm effectively disambiguates words with many possible senses by **using positive and negative samples in fine-tuning**, helping it learn subtle distinctions and leverage rich contextual knowledge. Similarly, the NB-based Lesk's Algorithm performs well through TF-IDF-based **probabilistic scoring**, focusing on relevant terms and adapting to nuanced sense variations, achieving the highest accuracy among all methods tested.

► ***Hyper-parameter Tuning.*** In this study, hyper-parameter tuning was limited to **one key parameter** in each proposed model to optimize performance without extensive modifications. For the BERT-based Lesk's Algorithm, the parameter adjusted is the number of **fine-tuning epochs**, with results presented in Tab. 1b. For the NB-based Lesk's Algorithm, tuning focuses on the **smoothing factor alpha**, with performance metrics shown in Tab. 1c.

4 Discussion

► ***Observation of Baseline Methods.*** From the experiment results, **MFS significantly outperforms the traditional Lesk's algorithm** in the accuracy metric. This observation could be attributed to two main reasons. First, MFS relies on **WordNet's predefined sense order**, where the most common sense typically aligns with general language usage, providing a strong **default prediction**. Second, traditional Lesk's Algorithm depends on **lexical overlap** between context and sense definitions, which can be limited by the **sparse vocabulary** in definitions; this often results in inadequate overlap, reducing Lesk's effectiveness in accurately identifying the intended sense.

► ***Limitation and Failure Analysis.*** The **BERT-based method**, while leveraging powerful **contextual embeddings**, struggles with highly **ambiguous contexts** or sentences lacking clear **disambiguation cues**. When context does not sufficiently distinguish between possible senses, BERT may align the ambiguous word with a similar but incorrect sense due to close **cosine similarity scores**. This issue is evident in Tab.2, where BERT misinterprets "**washington**" as the state rather than the U.S. capital due to ambiguous context. Besides, the **NB-based method**, limited by its assumption of **feature independence**, fails to capture nuanced interactions between context words. For example, in Tab.2, it incorrectly selects "action.n.01" over the correct sense "action.n.09" due to its reliance on isolated word probabilities, overlooking the broader context. Additionally, the **small test set size** and **inconsistent quality** limit the evaluation, as the limited number of relevant test samples for certain words introduces **bias in accuracy measurement**. When only a few categories represent a word's possible senses, the **model may appear to perform well or poorly based on chance rather than true capability**, skewing the results. This reduced sample diversity makes it challenging to gauge each model's generalizability across a broader range of senses and contexts.

References

[1] Explosion AI. spacy english models documentation, 2023. URL https://spacy.io/models/en. Accessed: 2024-11-11.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

[3] 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*, pages 24–26, 1986.

[4] Roberto Navigli, David Jurgens, and Daniele Vannella. Semeval-2013 task 12: Multilingual 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)*, pages 222–231, 2013.