Programming Assignment 2: Word Sense Disambiguation

Preprocessing and Evaluation Metrics:

Preprocessing:
The data was preprocessed through tokenization and lemmatization to ensure consistency with WordNet lemmas. Stop word removal focused on meaningful content words for overlap computation in Lesk-based methods. Multi-word expressions were standardized (e.g., using underscores) to match WordNet's format.

Evaluation Metrics:
Accuracy was used as the primary metric, considering predictions correct if they matched any gold-standard sense. A mapping between lemma sense keys in the gold standard and WordNet synsets was implemented for accurate evaluation.

Baseline Methods: MFS Baseline and Lesk Algorithm

Model Descriptions:
The Most Frequent Sense (MFS) baseline predicts the most observed sense of a word from training data, achieving **67.52% accuracy on the development set** and **61.72% on the test set**. It is computationally efficient but fails to account for context, limiting its performance with ambiguous cases. The Lesk algorithm improves upon this by using WordNet glosses to identify senses based on the overlap between glosses and the context of the target word. Lesk achieved **33% accuracy on the development set** and **33.6% on the test set**, showing limited effectiveness due to its dependence on gloss quality.

Successes and Difficulties Faced:
The MFS baseline excelled in identifying frequent senses where contextual clues were unnecessary. However, its context-agnostic nature made it unreliable for polysemous words. The Lesk algorithm incorporated context but struggled with short sentences and computational complexity from processing multiple glosses for each candidate sense.

Analysis and Suggestions for Improvement:
To enhance MFS, incorporating context-sensitive embeddings (e.g., BERT) could improve performance for ambiguous words. Lesk could benefit from richer glosses using external resources or semantic similarity measures instead of simple overlap metrics, making it more scalable and adaptable to diverse contexts.

Additional Experiment: BERT and Improved BERT

Model Descriptions:
The BERT model utilized contextual embeddings to compare similarity between the context sentence and sense definitions, achieving **52.06% accuracy on the development set** and **46.62% on the test set**. Improved BERT further enhanced performance by explicitly extracting embeddings for the target word and enriching sense representations using combined definitions and examples from **WordNet**. This approach achieved **57.73% accuracy on the development set** and **55.24% on the test set**.

Successes and Difficulties Faced:
BERT effectively leveraged contextual understanding for common senses but struggled with subtle distinctions and polysemy due to limited sense representation. Improved BERT addressed these issues by focusing on target-word-specific embeddings, yielding higher accuracy. Both models, however, were computationally expensive and sensitive to the quality of WordNet glosses.

Analysis and Suggestions for Improvement:
Fine-tuning BERT on domain-specific corpora could address issues of insufficient glosses. Leveraging external

resources, such as Wiktionary, or incorporating LLM-based definitions might enrich sense representations. Computational overhead could be reduced by adopting lightweight embeddings or knowledge distillation techniques, improving scalability for larger datasets.

Additional Experiment: Enhanced Lesk and Yarowsky Algorithm

Model Descriptions:
The Enhanced Lesk algorithm broadened the context for disambiguation by including definitions from related synsets (e.g., hypernyms, hyponyms). It achieved **40.7% accuracy on the development set** and **46.2% on the test set**. The Yarowsky algorithm utilized semi-supervised learning to propagate sense assignments iteratively, with **hyperparameter tuning** optimizing seeds, iterations, and context window size. Yarowsky achieved **14.43% accuracy on the development set** and **14.28% on the test set**.

Successes and Difficulties Faced:
Enhanced Lesk outperformed standard Lesk by leveraging hierarchical relationships within WordNet but faced computational challenges and sparse gloss coverage. Yarowsky effectively utilized minimal labeled data but required computationally expensive hyperparameter tuning and often struggled with convergence.

Analysis and Suggestions for Improvement:
Enhanced Lesk could integrate embeddings for similarity computations, could benefit from external lexical resources, such as Wikipedia reducing dependence on manual glosses. Yarowsky could be improved by automating hyperparameter tuning using Bayesian optimization and enhancing seed initialization with pre-trained language models. These refinements would address scalability and performance variability.

Sample output: (For Single instance: dict_items([('d001.s001.t002', ['group%1:03:00::'])]))

Single Instance MFS Prediction: {'d001.s001.t002': 'group%1:03:00::'}

Single Instance Lesk Prediction: {'d001.s001.t002': 'group%2:33:00::'}

Single Instance enhanced_lesk Prediction: {'d001.s001.t002': 'group%1:03:00::'}

Single Instance yarowsky_algorithm Prediction: {'d001.s001.t002': 'group%1:03:00::'}

Results:

Experiment	Dev set accuracy	Test set accuracy
MFS Baseline	67.52%	61.72%
LESK	33%	33.6%
BERT	52.06%	46.62%
Improved BERT	57.73%	55.24%
Enhanced LESK	0.41	0.46
Yarowsky’s	14.43 %	14.28%

This report presents a comprehensive evaluation of various methods for word sense disambiguation, ranging from baseline approaches like Most Frequent Sense (MFS) to advanced models such as BERT, Improved BERT, Enhanced Lesk, and Yarowsky algorithms. The experiments highlight the strengths of heuristic methods like MFS in simplicity and efficiency, while advanced models demonstrated superior accuracy through contextual understanding and enriched representations. However, challenges such as computational overhead, reliance on gloss quality, and limited scalability were identified across all methods. Proposed improvements, including leveraging external lexical resources, embeddings, and automated optimization techniques, suggest promising directions for enhancing model performance and scalability in future work. Overall, the experiments underscore the importance of balancing simplicity, contextual awareness, and computational efficiency in word sense disambiguation tasks.