Word Sense Disambiguation

Shwetali Shimangaud

1 Problem Setup

1.1 Problem Statement

This research aims to explore and compare various Word Sense Disambiguation (WSD) algorithms on the SemEval 2013 Shared Task #12 dataset using NLTK's WordNet v3.0 interface. The study evaluates the accuracy of two established methods, Lesk's algorithm and a most frequent sense baseline, alongside newer approaches employing bootstrapping and a pretrained BERT model. Challenges encompass handling lemma sense keys to synset numbers conversion, managing multi-word phrases, and accurately dividing instances into development and test sets, impacting the efficacy of WSD algorithms for at least five lexical items.

1.2 Input & Output

The problem statement utilizes the SemEval 2013 dataset with its corresponding golden key. Additionally, the bootstrapping method generates extra lexical resources with the assistance of ChatGPT. The system's output entails predicting the Word Sense for a provided context.

2 Experiments with results

2.1 Most Frequent Sense baseline

The baseline method, using the most frequent sense marked as #1 in the WordNet synset, attained an accuracy of **61.34%** on Dev Instances and **56.07%** on Test Instances. This simple frequency-driven approach established a fundamental starting point for comparison. The sample is shown in Figure 1a

2.2 Lesk

In the subsequent phase, an assessment was conducted on the NLTK Lesk's algorithm. The algorithm displayed a restricted accuracy of **29.38%** on the Dev Instances. Consequently, in an attempt to enhance the experiment, stop words were eliminated from the context, resulting in a decreased accuracy of **25.77%**. Additionally, varying the context window size within the Lesk algorithm experiments led to a decline in accuracy to **24%**. The superiority of the most frequent sense baseline over Lesk's algorithm on both Dev and Test Instances highlights the effectiveness of a simple frequency-based approach. Despite attempts to refine Lesk's algorithm by removing stop words, it remains constrained, emphasizing the intricacy of word sense disambiguation. Notably, these outcomes underscore the potential for improvement, signaling the need for exploring alternative methods. The sample WSD is shown in Figure 1b

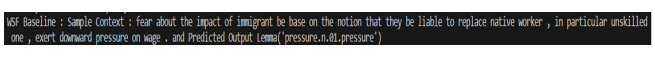
2.3 Bootstrapping

A bootstrapping technique was implemented targeting specific words: "**pressure**," "**group**," "**burden**," "**game**," and "**period**." The senses for each word are mentioned in Table 1Preprocessing involved standard procedures such as lowercase conversion, punctuation removal, stop word elimination, and lemmatization. The iterative WSD process utilized a Naive Bayes classifier.

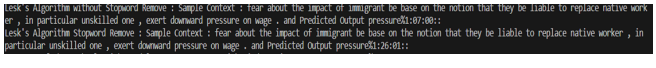
Iterative Refinement : Instances with the highest predicted probabilities were selected and included in the training set.

Per-Word Bootstrapping Experiments : Individual experiments were conducted for each target word, showcasing varying accuracies across iterations.

The effectiveness of the iterative process was visualized through the accuracy plot Figure 2a , illustrating the model's accuracy during training across iterations.Furthermore, the accuracies on the SemEval 2013 dataset's dev and test instances for each word are detailed in Table 2 , demonstrating the impact of iterative refinement on disambiguation accuracy. here as both dev and test instances were not having enough instances for each sense, the accuracy is 0.



(a) Most Frequent Sense baseline sample



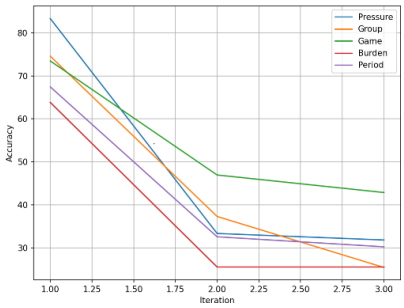
(b) Lesk sample

Word	Senses
Pressure	Influence, Measurement unit, Physical force
Burden	Load, Literary work, weight
Game	Contest with rules, Single play of sport, Secret scheme
Period	Amount of time, Interval taken, Geological time
Group	Unit, Chemical group, Mathematical group, Verb

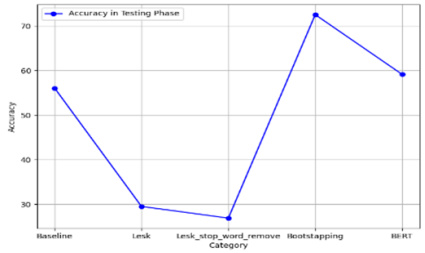
Table 1: Word Senses under consideration

Word	Accuracy on dev instances	Accuracy on test instances
Pressure	100%	75%
Burden	0	33.33%
Game	0	74.1%
Period	100%	0
Group	0	33.33%

Table 2: Accuracy of Bootstrapping



(a) Training accuracy for different words



(b) Accuracy over different models

2.4 BERT

In the final experimental phase, the implementation involved a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model. BERT, renowned for its capacity to be fine-tuned for diverse natural language processing tasks, including word sense prediction, exhibited an overall accuracy similar to the Baseline model. While the BERT model offers the advantage of pre-training, its limitation lies in the fixed context window size. Moreover, fine-tuning the BERT model proves to be computationally demanding.

3 Discussion and Conclusion

In conclusion, the investigation revealed distinct performances among various Word Sense Disambiguation (WSD) approaches. The most frequent sense baseline model demonstrated a stable performance of approximately 62%, whereas Lesk's algorithm, even after meticulous preprocessing and context window considerations, yielded unsatisfactory results.
The application of the bootstrapping method, relying on '**high confidence prediction**' as training seeds, showcased higher accuracy. However, an intriguing observation emerged: as the number of iterations increased, the model's performance diminished. This decline was attributed to overtraining, primarily due to the Naïve Bayes model's susceptibility to overfitting combined with the limited dataset.
Lastly, the utilization of the pre-trained BERT model showcased comparable performance to the baseline model. Despite its advantage in pre-training, the BERT model's fixed context window size and computational demands were notable limitations. The accuracy for each model is shown in Figure 2b
These findings underscore the challenges inherent in WSD tasks, emphasizing the need for tailored approaches to mitigate issues like overfitting with limited datasets. While certain methods exhibited promise, further exploration and refinement are crucial for advancing the accuracy and efficacy of WSD models.

4 Limitations

Creating the sample dataset posed a challenge within the bootstrapping method. Obtaining a dataset encompassing various senses of a word required careful consideration. The use of sentences generated by OpenAI, limited in vocabulary, may potentially lead to suboptimal performance when applied to real-time data. Additionally, the Lesk algorithm exhibits high sensitivity towards Polysemy, facing issues related to sense overlap.

5 Suggestions

To refine Word Sense Disambiguation (WSD) algorithms, incorporating Knowledge Graphs offers a potent strategy. By integrating semantic relationships between words and senses within the graph structure, we can enhance contextual understanding crucial for disambiguation. Strategies include leveraging graph embeddings, expanding contextual insights, utilizing graph-based algorithms, and adapting dynamic graph structures. This integration taps into the rich semantic associations present in Knowledge Graphs to significantly elevate disambiguation accuracy.