COMP 550: Word Sense Disambiguation

Sienna Hsu 260948832

I. Introduction

In this project, I employed 4 different models to disambiguate the senses of words given by a 1450-instance test dataset. The 4 models are the baseline model, Lesk's Algorithm model, the semi-supervised Label Spreading (LS) model, and the Complement Naive Bayes (CNB) model. The first two models were evaluated on the whole test dataset; whereas the final two models were evaluated on 5 chosen words in the test dataset. The baseline model achieved 59.8% accuracy, while Lesk's model reached only 34.8%. The LS and the CNB models showcased varied accuracies, depending largely on the test cases available in the test dataset and the text pre-processing choices.

II. Dataset

A development set of 194 instances and a test set of 1450 instances were available. The test data were lowercased, lemmatized, and removed of stop words and punctuation for the baseline and Lesk's models. To train the LS and the CNB models, I first selected 5 words that appeared in both development and test sets and are among the most frequent words in the test set. These five words are "country", "game", "action", "claim", and "time". Each word was disambiguated between two senses. There are one LS model and one CNB model built specifically for each word. To obtain more training instances for each word, I asked ChatGPT-3.5 to generate 50 sentences containing the target word with sense 1, 50 sentences with sense 2, and 100 sentences with sense 1 and sense 2 arbitrarily. Please see prompts.txt for the prompts used. I thus obtained 100 labeled training instances and 100 unlabeled training instances. The labeled instances in the development set are added to the labeled training set. The training data were lowercased, cleared of stop words and punctuation, lemmatized, tokenized into uni-, bi-, and tri-grams, and count-vectorized.

III. Experiments and Results

A. Baseline Model

The baseline model merely retrieved the most frequent word sense from WordNet. It relied on the words in the test dataset being used in the most common way. The model achieved 59.8% accuracy.

B. Lesk's Algorithm Model

Given a target word, Lesk's model calculated the number of overlapping words between the target word's context and each of its Synset definitions. The model predicted that the sense is the one with the most overlapping words with the target word's context. This approach performed poorly with a 34.8% accuracy. This might be due to several reasons: (1) The given contexts/sentences were not indicative of the word sense or were too short. (2) Lesk's algorithm does not consider the probability of each sense occurring when, in reality, some senses occur more commonly. (3) Related word senses and their definitions could naturally have some overlapping words, which could confuse the model.

C. Semi-Supervised Label Spreading Model

For the third model, I used the semi-supervised Label Spreading model from Scikit-learn. The self-training part is based on Yarowsky's algorithm. The classifiers/kernels used were K-Nearest Neighbors (KNN) and Radial Basis Function kernel Support Vector Machine (RBF). The LS model utilized both the labeled data and unlabeled data. For each of the five words, I tuned the hyperparameters *kernel*, *gamma*, *n_neighbors*, and *alpha* for the uni-, bi-, and tri-gram models and adopted the best-performing models. In total, for five words and three pre-processing choices, there were 15 models. Some models did better with KNN, and some with RBF. Please see the source code for the details on the hyperparameters chosen. The models' performance is summarized in table I.

TABLE I
Label Spreading models' performance (accuracy%)

Word \ Model	unigram	bigram	trigram
country	94	100	100
game	74	9	0
action	100	100	100
claim	89	89	100
time	67	33	67

D. Complement Naive Bayes Model

The final model is Complement Naive Bayes from Scikit-learn. I chose a Naive Bayes model due to the small labeled training dataset. Additionally, I decided to use Complement NB over Multinomial NB as the former has been shown to outperform the latter in text classification. The CNB model worked with uni-, bi-, and tri-gram data. As with LS models, the 15 CNB models underwent hyperparameter-tuning. For all CNB models, little to no smoothing turned out to work better. Additionally, specifying the class prior to [0.5, 0.5] improved the performance. (Although after adding the few instances available in the development set to the ChatGPT-generated dataset, the true distributions of the training datasets were not [0.5, 0.5], it was not far off.) The performance of the CNB model is summarized in table II.

TABLE II
CNB MODELS' PERFORMANCE (ACCURACY%)

Word \ Model	unigram	bigram	trigram
country	89	100	100
game	91	26	0
action	55	100	100
claim	89	89	100
time	50	67	67

IV. ANALYSIS

A. N-gram's Effect

I experimented with unigram, bigram, and trigram because the context immediately surrounding the target word could be important in word sense disambiguation. This was confirmed by the general trend of higher accuracies coming with higher degrees of n, as shown in tables I and II. The major outlier is the word "game." The test instances of "game" are mostly about soccer games, while the training data does not particularly focus on soccer. As it turned out, when grouping contexts into bigrams or trigrams, the contexts pointed "game" more towards "secret plan" as opposed to "a single play of a sport or other contest".

B. Comparing All Models

Table III displays the best performance of each of the 4 models. Lesk's model performed poorly for reasons discussed in section III (B). The baseline model's results depended heavily on the test data. It happened that both "country" and "action" only appeared in the test data in their most common senses, so the baseline model has a 100% accuracy for both words. The improvement in accuracies by using machine learning models like LS and CNB showed that there could be patterns not captured by rule-based methods like Lesk's algorithm. By merely tokenizing the context and counting the frequencies of each token, LS and CNB were able to more accurately disambiguate between two senses. Lastly, although LS has double the size of training data compared to CNB, CNB performed better. LS's strength might shine through more if the labeled data size is smaller.

TABLE III
COMPARISON OF DIFFERENT MODELS' PERFORMANCE (ACCURACY%)

Word \ Model	Baseline	Lesk's	LS	CNB
All words	59.8	34.8	-	-
country	100	5.6	100	100
game	0	0	74	91
action	100	0	100	100
claim	10	0	100	100
time	66.7	33.3	67	67

C. Sample Output

Here we take a look at a test instance of "time". The two senses for time are%1:11:00:: and %1:28:05::. The context of the instance is "some lawyer say appeals_court review Mr. Hayes 's conviction half a century later might be intrigue by the argument that new rule could be necessary to govern way of communicate from courtroom unimagin at the time of Dr. Sheppard 's case ." The gold standard sense for this instance is %1:11:00::, the most common sense for "time". Baseline model naturally outputted the sense correctly. The LS trigram model and the CNB bigram and trigram models also correctly predicted the sense. Lesk's algorithm did not output the correct sense. The context has more overlapping words with the definition of "time" as a resource as apposed to "time" as an event.

V. DISCUSSION AND CONCLUSION

We should look at the results of the experiments with caution because the test sizes of words range from 6 to 23. Additionally, because the given development set was small, I had to augment training data with ChatGPT, which often generates sentences of the same form. Nevertheless, we can conclude that machine learning models outperform chance-based baseline model and rule-based Lesk's model. Further investigation is required where both the training and test datasets are more

carefully curated and more than two senses are to be disambiguate.

REFERENCES

Overleaf is absolutely not letting me cite the sources correctly and I have been trying for more than an hour. Please see references.txt for citations. I apologize for this.