Word Sense Disambiguation
Sagar Nandeshwar - Student ID: 260920948

Objective:

In this project, I implemented 4 different Word Sense Disambiguation methods to determine the sense (meaning) of the word in the given context and compared their accuracy.

Dataset:

- Dev and Test Instance: subset of **SemEval 2013 Shared Task #12 (Navigli and Jurgens, 2013) dataset.**[1]
- **Boot Dataset:** I have generated 10 sample sentences using large **T5 model[4]** for each of the lemma in dev and test instances.
- **WordNet v3.0** for lexical resource

Preprocessing:

- **Lemmatization:** Reduces words to their base or dictionary form, known as a lemma.
- **Remove Stop words:** Stop words do not add much significant meaning to a sentence.
- **Remove Punctuation**
- **Tokenization:** Splits a sentence into individual words (token).
- **POS tagging:** categorizing words in a text (corpus) in correspondence with a particular part of speech
- **Feature Vectors:** Convert a collection of text documents to a matrix of token counts

Models and Algorithms

Model 1: The most frequent sense baseline

Select the first in the synset according to WordNet for the lemma

Model 2: Lesk algorithm:[2]

Steps to disambiguate word w:

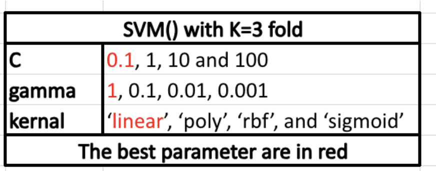
1. Construct a bag of words representation of the context, B
2. For each candidate sense Si of word w:

- Calculate a signature of the sense by taking all of the words in the dictionary definition of Si.
- Compute Overlap(B,signature(Si))

3. Select the sense with the highest overlap score

Model 3: Supervised Machine Learning with SVM:

The primary goal of SVM is to find a hyperplane (or decision boundary) that best divides a dataset into classes.
I have created an SVM model for every lemma present in dev instance and dataset Boot, trained them on dev instance and Boot dataset. I labeled the Boot dataset with both model 1 (most frequent synset) and model (Lesk algorithm). Lastly, I used a test instance to evaluate model performance.



SVM() with K=3 fold	
C	0.1, 1, 10 and 100
gamma	1, 0.1, 0.01, 0.001
kernal	'linear', 'poly', 'rbf', and 'sigmoid'
The best parameter are in red	

Model 4: Bootstrap Algorithm

I implemented modified Yarowsky's Algorithm[3] that was introduced in the lecture. I created Decision Tree Model for every lemma cover in the Dev instance and Boot dataset. The are three bootstrapping layers in the following model. For lemma in Boot dataset, that are not covered in seed set, I label them using model 1 or mode 2 with probability of 0.5 and give confidence score 0.6 and 0 .4 (their observed accuracies in this project), respectively.

(Seed Set)

1. Consider dev_set as a seed set and trained the first model

(Bootstrap Layer 1)

2. Apply the supervised model on Boot dataset.
3. Select the labels with prediction confidence of at least 0.9, and formed a new seed set
4. Trained the model on the new seed set

(Bootstrap Layer 2)

5. Apply the model again on Boot dataset.
6. Select the labels with prediction confidence of at least 0.5, and formed a new seed set

(Bootstrap Layer 3)

7. Trained the model on the new seed set

8. Apply the new model on the rest of the dataset
9. Used the label and trained the final model

(Evaluation)

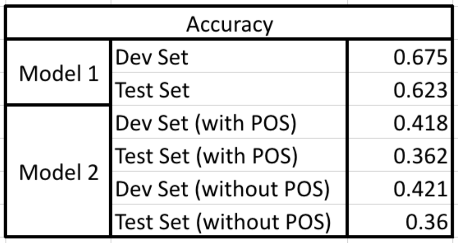
10. Apply the final model on the test instance for evaluation

Evaluation

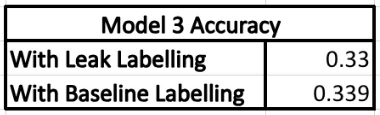
For the evaluation, I used the accuracy score of the models:

Accuracy = Number of correct predictions / Total number of predictions

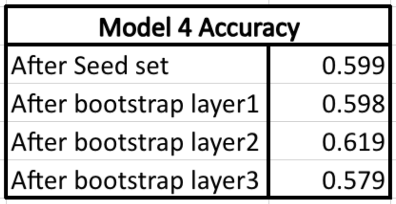
Results



Accuracy		
Model 1	Dev Set	0.675
	Test Set	0.623
Model 2	Dev Set (with POS)	0.418
	Test Set (with POS)	0.362
	Dev Set (without POS)	0.421
	Test Set (without POS)	0.36



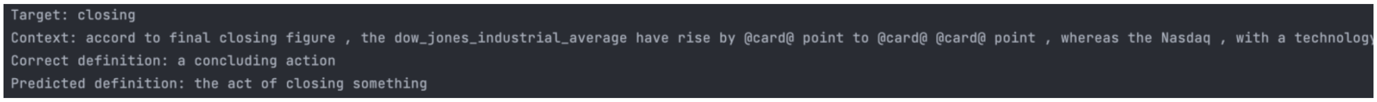
Model 3 Accuracy	
With Leak Labelling	0.33
With Baseline Labelling	0.339



Model 4 Accuracy	
After Seed set	0.599
After bootstrap layer1	0.598
After bootstrap layer2	0.619
After bootstrap layer3	0.579

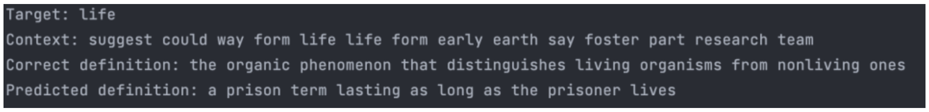
Sample Output

Model 1:



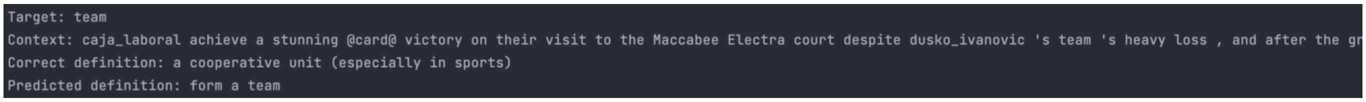
```
Target: closing
Context: accord to final closing figure , the dow_jones_industrial_average have rise by @card@ point to @card@ @card@ point , whereas the Nasdaq , with a technology
Correct definition: a concluding action
Predicted definition: the act of closing something
```

Model 2:



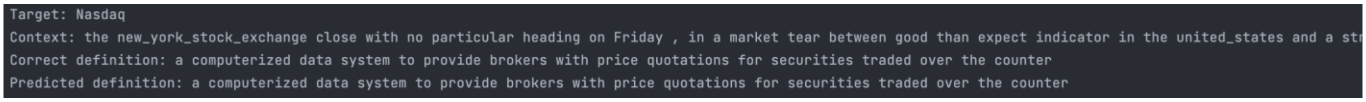
```
Target: life
Context: suggest could way form life life form early earth say foster part research team
Correct definition: the organic phenomenon that distinguishes living organisms from nonliving ones
Predicted definition: a prison term lasting as long as the prisoner lives
```

Model 3:



```
Target: team
Context: caja_laboral achieve a stunning @card@ victory on their visit to the Maccabee Electra court despite dusko_ivanovic 's team 's heavy loss , and after the gr
Correct definition: a cooperative unit (especially in sports)
Predicted definition: form a team
```

Model 4:



```
Target: Nasdaq
Context: the new_york_stock_exchange close with no particular heading on Friday , in a market tear between good than expect indicator in the united_states and a str
Correct definition: a computerized data system to provide brokers with price quotations for securities traded over the counter
Predicted definition: a computerized data system to provide brokers with price quotations for securities traded over the counter
```

Discussion and Conclusion:

Model 1 has performed significantly better than Model 2 on both dev instance and test instance with accuracy of 0.675 and 0.623, respectively. This may be due to the fact that sentences containing lemmas, where lemma of less frequent form is used, are rare. POS tagging does seem to have much impact on accuracy of model 2. Different labelling schemes for boot dataset label has no impact on Model 3 accuracy. Model 4 (Bootstrapping) has performed better than both the versions of the Model 3, as bootstrap does give more insight into dataset.

Improvement:

Model 3 and Model 4 can be improved with larger and more accurate datasets, which could be manually created by experts. We have seen that both model 1 and model 2, at best, have accuracy of 0.6 and 0.4 respectively, therefore the model trained on the dataset that are labeled by model 1 and model 2 will perform poorer. We could also implement a multi-feature ML model, with would significantly consume less space and would be more efficient. For Model 4, automating the bootstrap layers would help us to better fine tune the model.

Reference:

[1] "SemEval-2013 Task 12: Multilingual Word Sense Disambiguation" - Roberto Navigli Et al

[2] "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet" - Satanjeev Banerjee Et al

[3] "Analysis of Semi-Supervised Learning with the Yarowsky Algorithm" - Gholam Reza Haffari, Et al

[4] "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation" - Yusong Wu Et al