Sentiment Analysis
Feature addition and Accuracy improvement

Prof. Brian Reese, Akshina Banerjee

University of Minnesota – Twin Cities, College of Liberal Arts, Institute of Linguistics

Introduction

❖ **Definition:**

Sentiment analysis refers to the use of *natural language processing*, *text analysis and computational linguistics* to identify and extract subjective information in source materials. In simpler terms, it is a tool built in computer software to determine the opinion of the writer of a piece of text. Such texts generally either bear a positive, negative or neutral mood and that is what sentiment analysis seeks to find.

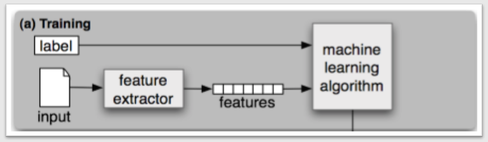
For example,

1. "The value of X company's shares skyrocketed" – Positive Sentiment

2. "This movie is the worst that I have seen in years" – Negative Sentiment

3. "The product was not the best but it suited well with some of my requirements." – Neutral Sentiment

❖ **Text Classification:**



❖ **Traditional Method of Text Representation:**

Bag of Words (BoW) approach

- ***What it is*** **:** A set of words that is chosen before the text classification. The selection can be made in multiple ways, e.g. it could be the n most frequent words in the entire training corpus.
- ***How it is used*** **:** The words from the text are matched to the existing words (and the sentiments that they denote) in BoW and then the classifier gives a prediction of the sentiment.
- ***Why it is used*** **:** Since this approach does not involve the employment of any linguistic structure, it is simple and this simplicity makes BoW popular
- ***Example*** **:** Classification of '**The movie was great**' (extraction of nouns and adjectives):

BoW = {movie, film, great, horrible, tedious} ⟶ Text representation = {movie:1, film:0, great:1, horrible:0, tedious:0}. This information is passed on to the training algorithm which will be trained to associate individual features (e.g great:1) with sentiment labels – in this case "positive"

❖ **Criticism for BoW :**

No linguistic structure is considered while classification because two main assumptions of BoW are – (a) Word order/ word position does NOT matter, (b) Lexical category of words (nouns vs verbs vs adjectives etc) does NOT matter.

❖ **Illustration of Problem:**

Classification of '**The company never failed**' (extraction of nouns and verbs):

Let BoW = { company, failed } ⟶ Text representation = { company: 1, failed: 1, succeded: 0 }

Sentiment Label = "Negative" ⟶ **WRONG SENTIMENT LABEL!**

Research Question

❖ **Research Question:**

The previous illustration revealed the following shortcoming of the BoW model:

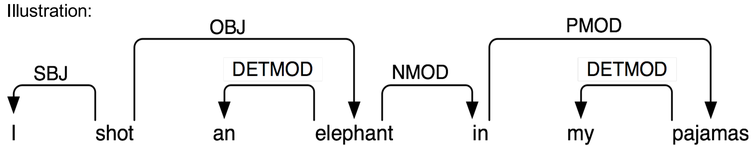
Information on **dependency** is excluded. Dependency refers to how a word **modifies** another and contributes to a **meaning shift** of the modified word.

Example: 'never' modifies 'failed' to mean situations of success or neither success nor failure.

Thus, this research seeks to **ADD** the feature of **dependencies** to the traditional method of text classification to check whether accuracy rates of sentiment analysis improves.

So as to not exclude any lexical item, all the words in a given text will be extracted, along with each dependency.

Illustration:



Methodology

- **Data used:** IMDB movie data set
- **Total number of movie reviews:** 42929
- **Steps taken to add feature and calculate accuracy score:**

❖ Step 1:

Movie review with sentiment ⟶ (Dependency Parser) ⟶ Movie review representation: dependency parsed

Example: Dependent: 'never'; Governor: 'failed'

❖ Step 2:

Movie review representation: dependency parsed ⟶ (Feature Extraction) ⟶ Formation of dependency pairs

Example: dependent + governor : 'never + failed'

❖ Step 3:

Movie review representation: Dependency pairs, individual Words ⟶ (Scikit Learn) ⟶ Naïve Bayes Classifier

❖ Step 4:

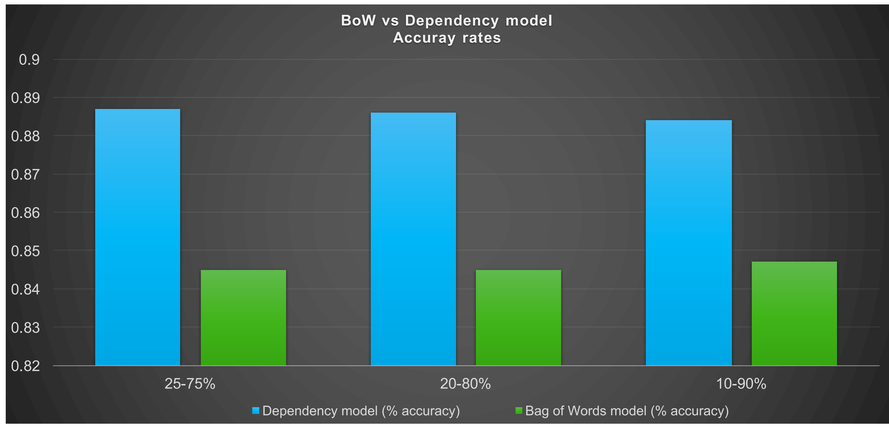
Classifier ⟶ (Test-Train Split) ⟶ Accuracy rates

Data / Observations

Train-Test split	Dependency model (accuracy)
25-75%	0.887
20-80%	0.886
10-90%	0.884

Results and Conclusion

❖ **Significance testing:**



- The p-value for the difference in the accuracy rates under all the three splits is **less than 0.00001**.
- Since α= 0.05 was chosen, any p-value below 0.05 indicates that the results are **significant**.

❖ **10-fold cross validation :**

Dependency Model

Cross Validation:	Scores:
1	0.877
2	0.842
3	0.836
4	0.850
5	0.857
6	0.854
7	0.839
8	0.838
9	0.860
10	0.828

BoW Model

Cross Validation:	Scores:
1	0.843
2	0.849
3	0.852
4	0.853
5	0.849
6	0.844
7	0.858
8	0.844
9	0.848
10	0.845

- The observations under the cross validation are puzzling because the deviations from the mean accuracy score are high.
- The same cross validation for the BoW model has very small deviations in accuracy scores, if any.

❖ **Conclusion:**

- Adding the feature of dependencies significantly improved the accuracy rates.
- Further research should look into:
  - Why the cross validation is showing an anomalous behavior in case of the dependency model.
  - Tokenizing the corpus for html tags and re-running the experiment.
  - The most informative features to see how the dependency pairs are classified.

Acknowledgments and Selected References

Acknowledgments:

- I would like to thank my mentor Professor Brian Reese for immense support and active involvement throughout the research project.
- I would like to thank a fellow student and friend Aaron Free for his massive contribution towards the computational part of the project.
- Last but not the least, I would like to thank Undergraduate Research Opportunity Program (UROP) for funding this project.

Selected References:

- Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." *Foundations and trends in information retrieval* 2.1-2 (2008): 1-135.
- Maas, Andrew L., et al. "Learning word vectors for sentiment analysis." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011.