Rhetorical Figure Detection in Political Texts

Zubair Rafiq, Ahmad Bilal Sohail, Eltun Ibrahimov
Faculty of Computer Science and Mathematics
University of Passau, Germany
{rafiq01,sohail01,ibrahi10}@ads.uni-passau.de

1 INTRODUCTION

Rhetorical figures can be described as a word or a phrase used in the non-literal sense so that the speech/writing becomes more persuasive, relatable and vivid. The literal meaning of the phrase or word is not taken into account. For example “He racked his brain so that he could come up for new ideas for his book”. Generally, the literal meaning of rack is a device or instrument that in earlier ages people used to torture the servant or to break their limbs apart. But here it is used to make a great effort to think or to remember something. The figures of repetitions are a family of figures. They usually involve repetition of any linguistic element, ranging from sound, as in rhyme, to concept and ideas, as in tautology and pleonasm. A computer can easily detect the repetition of words on the other hand detecting only the ones provoking a rhetorical effect requires much effort and the reason is the presence of many irrelevant and accidental repetitions.

1.1 Motivation

We believe that this is a challenging project which is going to offer us great learning skills in terms of speech exploration and analysis, data pre-processing, data exploration, machine learning and visualization techniques. After successful completion of this project, we will be able to differentiate between a good and a persuasive speech. The reason we chose the following mentioned figures of speech is that they make their extraction from the text pretty straightforward. It needs an algorithm for locating repetition cases and since we are not specialized nor have any help from any specialist in the linguistic area, these figures were the most manageable for us to annotate. Working with these figures lets us concentrate more on the machine learning phase of the project.

1.2 Goals

- A machine learning pipeline will be designed which will detect the potentially rhetorical figures from the Hansard Speeches corpus.

- In this project we will focus on the following three rhetorical devices of the repetitive figures family: Epistrophe, Epanaphora, and Epanalepsis.

  – Epistrophe: repetition of the same word or words at the end (or near the end) of successive phrases, clauses or sentences. Also known as epiphora or antistrophe. (Ali wants pizza, Ahmad wants pizza, in fact, everybody wants pizza)

  – Epanaphora: repetition of the same word or group of words are repeated at the beginning of two or more clauses or sentences. Also known as Anaphora. (I can act. I can sing. I can do whatever I want to do)

  – Epanalepsis: repetition of similar words/phrases at the beginning and end of the same sentence. (A poor can feel the pain of a poor)

- The main challenge here is to detect only truly provoking rhetorical figures as there will be many false positive cases of repetitions as well which makes it a needle in the haystack.

2 PLAN

(1) A corpus of United States Presidential Speeches[1] will be used which contains speeches made by

[1] https://www.kaggle.com/littleotter/united-states-presidential-speeches

American politicians between 1789 to 2019.

(2) At the very next step, the data will be cleaned and pre-processed because the dataset contains the columns like name of the presidents, political affiliation of the presidents and transcripts. Only the selected columns will be used to perform the task.

(3) An algorithm will be designed for extracting all the candidates for the chosen rhetorical figures by comparing the word repetitions and using the selected features for different figures.

(4) Decision Tree classifier will be used to train on 80% of data which will then be tested of the rest of 20%.

(5) F1 scores look promising for the evaluation phase.

(6) We will be using pandas, nltk, scikit-learn, re and matplotlib libraries of python.

3 RELATED WORK

This work is inspired from the previous work done on Presidential Speeches in 'An Annotation Tool for Automatically Detecting Rhetorical Figures' (Gawryjolek *et al.*, 2009). We also highly benifited fron the research done by Marie Dubremetz and Joakim Nivre on these two figures of speech. In the paper of 'Rhetorical Figure Detection: Chiasmus, Epanaphora, Epiphora' (Dubremetz and Nivre, 2018) they focus on the figures which we have selected to work on. Different from our approach to the task, in their project they treat it as a ranking task, using a linear classifier that outputs a certain floating point number used for ranking the cases from the most obvious to the accidental occurrence of repetition. They set weights for the features and adjust them manually until they achieve the desired result by the machine. The result of the application is a sorted descending list regarding the relevance of the figure. For annotation of the corpus are used two specialized linguists which have to both agree for a figure in order to classify it as a true case.

Their other paper which we considered is 'Rhetorical Figure Detection: the Case of Chiasmus' (Dubremetz and Nivre, 2015) and 'Machine Learning for Rhetorical Figure Detection: More Chiasmus with Less Annotation' (Dubremetz and Nivre, 2017)

We also considered the following papers: 'Automated Annotation and Visualization of Rhetorical Figures' (Gawryjolek, 2009) and 'Rhetorical Figure Annotation with XML' (Ruan *et al.*, 2016)

4 DATA PREPROCESSING

As discussed in Section 2, the used corpus contains more information, thus it is required to extract only some columns and preprocess the data. In the very first step, the required column containing speeches(Transcripts) should be extracted from the dataset.

As many real-world datasets may contain missing values and duplicates, the corpus should be cleaned out from them. Having this stage implemented, the resulting corpus contains approximately 4.2 million words and the speeches from 44 presidents of United States. Unfortunately, the number of speeches for each president is not fairly equal, making it difficult to make any comparisons regarding the frequency of the rhetorical figures usage between presidents.

In comparison with the other NLP tasks, some preprocessing stages should not be carried out before the detection of rhetorical figure candidates, since these algorithms utilize the punctuation and ordering.

4.1 Detection of rhetorical figure candidates

Use of rhetorical devices in political speeches makes them more meaningful and memorable. There exist a wide range of rhetorical figures utilized in both written and spoken cases. In this work, a small subset of rhetorical devices (i.e., epistrophe, epanaphora and epanalepsis) was chosen to analyse in detail. All these three figures are similar in nature while being relied upon word or phrase repetitions in different positions in a sentence [7].

Regardless of the chosen rhetorical figure, speech should be split in a multilevel way as shown in Figure 1.

(1) We implemented Regular Expressions to partition each speech to different sentences.
(2) Since rhetorical figures can also be a part of sentence, we split sentences into different parts based on the utilized punctuation marks (i.e., commas,

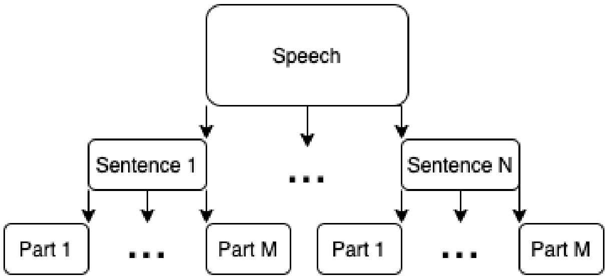

Figure 1: Structure of speech in different levels

semicolon and colon) by means of regular expression matching operations[2].

Having all these partitions(i.e., sentences and parts of sentences) as a sequence of words, they can be processed further as described in Algorithm 1 where SEQUENCES is a list of K sequences of words and REVERSE() is a function to reverse the content of a list.

Algorithm 1 Epistrophe, Epanaphora and Epanalepsis finder

```
LAST = SEQUENCES[1]
for $iteration = 2, 3, \dots K$ do
    $Epanaphora$ : Compute the first N common words of LAST and SEQUENCES[iteration]
    $Epistrophe$ : Compute the first N common words of REVERSE(LAST) and REVERSE(SEQUENCES[iteration])
    $Epanalepsis$ : Compute the first N common words of REVERSE(LAST) and SEQUENCES[iteration]
    Last = SEQUENCES[iteration]
end for
```

5 TASK DEFINITION

Our task starts with creating an algorithm for extracting all the cases of repetition. We begin with preprocessing the text by lowering the cases and removing numbers and punctuation, as these figures should contain only normal words. Then we create a list of tuples with the sentences we found and the features that we have decided to use for machine learning.

For Epanaphora we use these features:

(1) Sentence length : Rhetorical figures tend to be short. In long sentences Epanaphora loses the emotional effect it is intended to produce.

(2) Number of successive sentences starting with the same phrase : A repetition of the same phrase in more sentences is more likely to have been used intentionally rather than accidentally. For the extraction of the cases we search for minimally 3 consecutive sentences.

(3) Has strong punctuation : On most cases a rhetorical figure ends with a question mark or exclamation mark, so finding a special punctuation at the end increases the chances of it being an Epanaphora.

For Epistrophe we have chosen these features:

(1) Difference between sentences length : For a rhythmic and balanced sound of the rhetoric effect on the speech, sentences containing the Epistrophe should be similar in length. The smaller the difference the more the chances for being an Epistrophe.

(2) Number of successive sentences starting with the same phrase : Same logic as in Epanaphora's case.

(3) Is Identical : There are some cases of identical sentences on the corpus (clean your house!clean your house!clean your house!) or (don't do it.don't do it.don't do it), which is not qualified as Epistrophe, indeed these are another kind of rhetoric.

6 ML IMPLEMENTATION

Having all these features extracted, the correspoding classifier should be designed. Our task is to do binary classfication whether a text with the given parameters contains the specific rhetorical figure or not. Constructed feature matrices should be fed into the designed model with the respective labels. There exist many approaches to carry out such kind of classification. We decided to proceed with decision tree while it has some advantages over the other methods. For instance, if we compare the decision tree with the logistic

[2] https://docs.python.org/3/library/re.html

regression, two main merits can be considerably clear in our use cases.

(1) Decision tree algorithm continues iteratively with splitting the space into smaller regions, whereas in logistic regression single line is supposed to be computed to classify the space. Namely, decision tree, coming with the non-linear decision boundary, can be an ultimate choice for more satisfying outcomes.
(2) Decision tree presents the iterations which can be considered more interpretable and visual. In other words, algorithm can be visually described through a tree which shows the iterations of algorithm (see: Fig. 2 and 3). This functionality enables us to analyze and prioritize our feature matrix in detail.

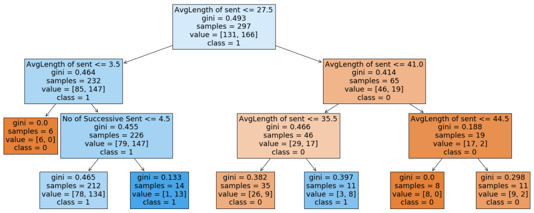

Figure 2: Decision tree for anaphora

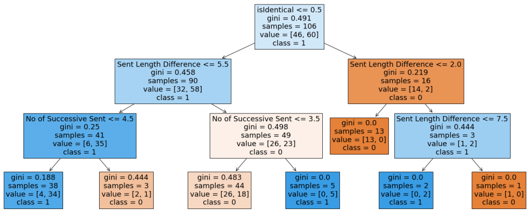

Figure 3: Decision tree for epistrophe

Although we considered and extracted strong punctuation as a feature in Section 5, as can be seen from Figure 2, it has a lower priority in decision-masking that it is not checked during the first three conditions. Whereas the average length of sentence is crucial since long sentences are expected to lose a sense of rhetoricality . Turning to the epistrophe (see: Fig. 3), it can be clearly shown than the feature checking an identicality of phrase plays an important role in the first partitioning. For both rhetorical figures, the number of successive sentences is equally important to decide and tended to be maximized. We implemented both classifier and validation metrics offered by [3] as a library.

7 RESULTS

In our 4.2 million words long corpus, our algorithm has detected 372 Anaphora candidates, 133 Epiphora candidates and only 57 Epanalepsis cases. Important here to understand that we can adjust our algorithms with different parameters, so it would yield more or less results of repetition. The configuration we use for sentence repetition number and words repetition number is the best for extracting a balanced dataset, so we find as many true cases as possible and reduce the number of overall repetitions that would have to be annotated.

The final score for Accuracy and F1-Measure is given in the table below:

	Epanaphora	Epistrophe
Accuracy	0.71	0.74
F1-Score	0.79	0.8

Table 1: Empirical results on Epanaphora and Epistrophe

8 CONCLUSION

At the end of this project we can evaluate our improved skills in this area of computer science. We have learned text pre-processing techniques, the art of feature engineering, data exploration and visualisation, selecting, training and predicting with a classifier, usage of many python libraries and modules.

We have failed to examine the useful features for Epanalepsis in order to use them for Machine Learning part for this figure. Therefore, more searching can be done for the figures by adjusting the parameters of the search such as number of words that should be repeated and number of consequent sentences. That would give a couple of real figures and many accidental cases of repetition. Then our time needed for annotation would

[3] https://scikit-learn.org/stable/

be increased drastically. The aspect of features selection can be further worked on. There should be other features more appropriate for describing these three figures. More testing on other corpora can be performed to double check the performance of the application.

As our personal conclusion we would say that this project was a success in terms of what we have learned and the vision that we have created for this area of the computer science. An important factor to be considered is the lack of specialized resources in the field of linguistics for helping us to annotate better and to select more expressing features.

REFERENCES

[1] J. Gawryjolek, C. DiMarco and R. Harris, An Annotation Tool for Automatically Detecting Rhetorical Figures SYSTEM DEMONSTRATION, 2009.

[2] M. Dubremetz and J. Nivre, Rhetorical Figure Detection: Chiasmus, Epanaphora, Epiphora, 2018, p. 10.

[3] M. Dubremetz and J. Nivre, Rhetorical Figure Detection: the Case of Chiasmus, Denver, Colorado, USA, 2015, pp. 23–31.

[4] M. Dubremetz and J. Nivre, Machine Learning for Rhetorical Figure Detection: More Chiasmus with Less Annotation, Gothenburg, Sweden, 2017, pp. 37–45.

[5] J. J. Gawryjolek, *Automated Annotation and Visualization of Rhetorical Figures*, 2009, http://hdl.handle.net/10012/4426.

[6] S. Ruan, C. D. Marco and R. A. Harris, Rhetorical Figure Annotation with XML, 2016.

[7] B. Englard, *A Rhetorical Analysis Approach to Natural Language Processing*, 2013.