Computer Science Department
University of Crete

Opinion Mining on Parliamentary Commentaries, using Machine Learning.

Moschonas Giannis, Smyrnaios Giorgos

Computer Science Department
UNIVERSITY OF CRETE
Heraklion, Greece 2015

Graduate thesis 2015

Opinion Mining on Parliamentary Commentaries, using Machine Learning.

Moschonas Giannis, Smyrnaios Giorgos

Computer Science Department
University of Crete

Computer Science Department
University of Crete
Heraklion, Greece 2015

Opinion Mining on Parliamentary
Commentaries, using
Machine Learning.

Supervisor: Name, Company or Department
Examiner: Name, Department

Graduate thesis 2015
Department of Computer Science
University of Crete
E-mail (Moschonas): jmoschon@csd.uoc.gr
E-mail (Smyrnaios): gsmyrneo@csd.uoc.gr

Typeset in LaTeX
Heraklion, Greece 2015

Moschonas Giannis, Smyrnaios Giorgos
Compute Science Department
University of Crete

Abstract

Natural Language Processing is a scientific field in the area of Computer Science, which seeks a better correlation between natural language and computers. In fact Natural Language Processing is a wide scientific field in which technologies such as “Machine Translation”, “Named Entity Recognition and Disambiguation”, “Sentiment Analysis” and more are included. This Thesis seeks a better approach in order to export information from plain texts, which basically contain civil placements on consultation laws issued by the Greek Government. Attempted to export proposals - counterproposal of the authors and also the arguments that the authors expressed. Finally attempted to export the entire view of the author summarized in a word “Positive” or “Negative”, according to the opinion that the author expressed in the text. To export of these data is made entirely by analysing texts through a three step process (which will be explained in detail in the following chapter of this Thesis) and implementing techniques from the wide spectrum of NLP (such as Information Retrieval, Part-Of-Speech Tagging, Sentiment Analysis, etc.). The results show that we can create realistic methods in order to export this type of Semantic Information. Recently the research community gives more interesting on this subject, because ic could be exploited in a number of other areas outside the field of Computer Science (eg. Journalism, Politics, etc.).

Keywords: argument extraction, sentiment, machine learning, suggestion extraction, POS Tagging, opinion mining, natural language processing.

Declaration of Authorship

We declare that this thesis titled, “Natural Languages Processing on Parliamentary Commentaries, using Machine Learning.” and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly at-tributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselfs jointly with others, we have made clear exactly what was done by others and what I have contributed ourselves.

Giannis Moschonas

Name	Sign

Giorgos Smyrnaios

Name	Sign

Acknowledgements

TODO

Name Familyname, Gothenburg, Month Year

Contents

List of Figures

List of Tables

1 Introduction

TODO

2

Background

TODO

2.1 Equation

$$f(t) = \begin{cases} 1, & t < 1 \\ t^2 & t \geq 1 \end{cases} \tag{2.1}$$

3
Methods

In this chapter, we will thoroughly analyse the ways with which the three processing stages which were presented in the previous units, were implemented.

In order to describe in the best possible way the process that was followed, a detailed description of the dataset which was used will be given, as well as of the features that characterise it. Then, we will describe the relational database model which we used to store the information from the texts (dataset). Finally, for each one of the process stages, we will describe the methodology with which each matter was approached.

3.1 Preparing the Dataset

In this part of the methodology, the features of the used dataset (3.1.1) will be described in detail. Some information on the way of choosing data (3.1.2) will be described as well. Next, the way of data mining from the Greek Open Government platform[1] (3.1.3) and finally, the Entity-Relation Model (3.1.4) of the database which was used to store the data will also be described.

3.1.1 Dataset

As it has already been mentioned in some points of this Thesis, the data that were used have been taken from the Greek Open Government platform, which constitutes a platform of electronic consultation of citizens on texts, more specifically on laws and decrees that the Greek Government issues. These data are open and accessible to everyone.

In this section, the basic features of the studied texts will be described. The reason why this section comes first in this part of the methodology, is that the very nature of these texts (they are basically users' comments to the online service), created many problems in their analysis.

As it has already been mentioned, the texts that were studied feature several oddities, some of which made the process of analysing them difficult.

[1] `http://www.opengov.gr`

- Initially, the first that we can notice is that the length of the texts is relatively short. To be precise, it is rare for them to be longer than 3000 characters (approximately 200 words, 80% of the texts). The length of the text did not affect all the stages of the analysis. The biggest difficulty appeared in the effort to extract the degree of the writer's agreement with the initial text (more details will be given later).

- A second remark is the fact that the texts that were studied do not consist an official text. By the term "official", we want first to declare that the texts are made up of users' comments in an online service and secondly, that they contain many errors (spelling etc). This created many difficulties in the studying of these comments. The first difficulty had to do with the tools needed in order to conduct the overall analysis of each text. The basic idea was that the tools had to be tolerant when it came to errors, at least up to a degree.

  Some very usual errors are:
  1. spelling errors
  2. absence of some letters in a word
  3. letter transposition in a word
  4. use only of capital letters
  5. absence of punctuation
  6. wrong sentence separation (there was no gap after the dot)
  7. some more errors that will not be mentioned for ease of reference

- One more issue is that there are many times when syntactic structure errors are spotted. This problem is directly connected to the use of POS Tagger for the syntactic analysis (parsing) of texts. This issue affects, to some degree, the extrapolation of arguments and of proposals and counter proposals that the user makes.

- Another feature is that the texts are entirely in Greek. This problem is more serious, because there are no tools which we needed at some point of the analysis, that support the Greek Language. Subsequently, as we will see later on, there was the need to resort to some compromising solutions.

- One last issue that is worth mentioning, which constitutes a more qualitative feature, at least in the whole of texts that were studied, is the fact that the majority of users who wrote a comment are "annoyed". This "annoyance" stems from the fact that the texts that are under discussion contain laws and presidential decrees that, essentially, lead to a decrease in public spending towards the citizens. This "annoyance" is noted almost in the entire dataset that we studied. The problem is that the texts in which the writers agree with the initial text are limited. As a result, this issue complicates the process of acknowledging, if the writer agrees with the initial text.

3.1.2 Choosing Set of Documents

TODO

3.1.3 Finalizing the Dataset

TODO

3.1.4 Database

TODO

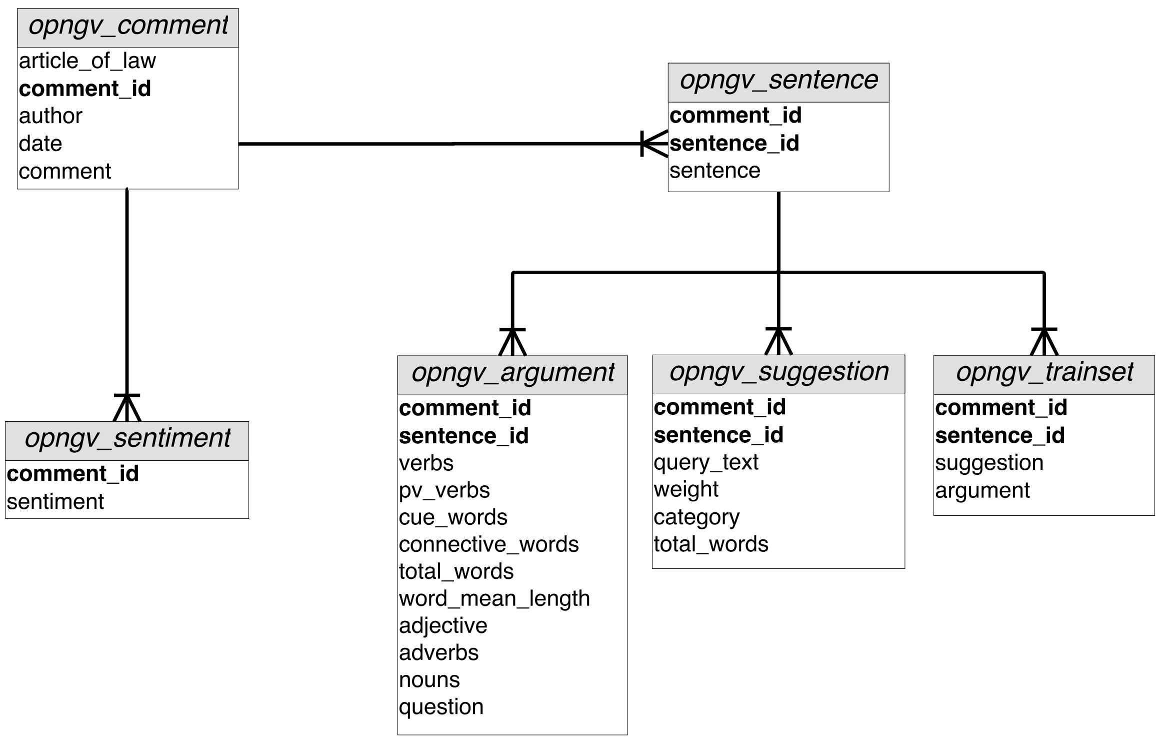

Figure 3.1: Entity - Relation Model

TODO

3.1.5 Building a Trainset

TODO

3.2 Argument Extraction

TODO

3.2.1 Selecting Argument Markers

TODO

3.2.2 POS Tagging

TODO

3.2.2.1 POS Tagger

TODO

3.2.2.2 POS Tagger Output

TODO

```xml
<?xml version='1.0' encoding='UTF-8'?>
<cesDoc xmlns="http://www.xces.org/schema/2003" version="0.4">
  <text>
    <body>
      <p id="p1">
        <s id="s1">
          <t id="t1" word="..." tag="AtDfNeSgNm" lemma="..."/>
          <t id="t2" word="..." tag="RgFwOr" lemma="..."/>
```

```
            <t id="t3" word="..." tag="PnReNe03SgNmXx" lemma="..."/>
            <t id="t4" word="..." tag="VbMnIdPr03SgXxIpPvXx" lemma="..."/>
            <t id="t5" word="..." tag="VbMnIdPr03SgXxIpPvXx" lemma="..."/>
            <t id="t6" word="..." tag="AsPpSp" lemma="..."/>
            <t id="t7" word="..." tag="NoCmFeSgAc" lemma="..."/>
            <t id="t8" word="..." tag="RgFwOr" lemma="..."/>
            <t id="t9" word="..." tag="PTERM_P" lemma="..."/>
          </s>
        </p>
      </body>
    </text>
</cesDoc>
```

TODO

3.2.2.3 Parsing XML File

TODO

3.2.2.4 Uploading to Database

TODO

```
INSERT INTO opngv_argument VALUES (values..)
```

3.2.3 Apply Machine Learning

TODO

3.2.3.1 Selecting Train and Test Set

TODO

```
SELECT
        opngv_argument.verbs,
        opngv_argument.pv_verbs,
        opngv_argument.cue_words,
        opngv_argument.connective_words,
        opngv_argument.total_words,
```

```
        opngv_argument.word_mean_length,
        opngv_argument.adjective,
        opngv_argument.adverbs,
        opngv_argument.noons,
        opngv_argument.question,
        opngv_trainset.Argument
FROM
        opngv_sentence
        INNER JOIN opngv_argument
                ON opngv_sentence.comment_id = opngv_argument.comment_id
                AND opngv_sentence.sentence_id = opngv_argument.sentence_id
        INNER JOIN opngv_trainset
                ON opngv_sentence.comment_id = opngv_trainset.comment_id
                AND opngv_sentence.sentence_id = opngv_trainset.sentence_id
```

TODO

3.2.3.2 Machine Learning Process

TODO

3.2.3.3 Machine Learning Algorithms

TODO

"Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes."

3.3 Suggestion Extraction

TODO

3.3.1 Selecting Suggestion Markers

TODO

```xml
<?xml version='1.0' encoding='UTF−8'?>
<cesDoc xmlns="http://www.xces.org/schema/2003" version="0.4">
  <text>
    <body>
      <p id="p1">
        <s id="s1" casing="lowercase">
          <t id="t1" word="..." tag="VbIsIdPr03SgXxIpAvXx" lemma="..."/>
          <t id="t2" word="..." tag="PtSj" lemma="..."/>
          <t id="t3" word="..." tag="VbMnIdXx03SgXxPePvXx" lemma="..."/>
          <t id="t4" word="..." tag="NoCmFeSgNm" lemma="..."/>
          <t id="t5" word="..." tag="AsPpPaFeSgAc" lemma="..."/>
          <t id="t6" word="..." tag="NoCmFeSgAc" lemma="..."/>
          <t id="t7" word="..." tag="DIG" lemma="..."/>
          <t id="t8" word="..." tag="AtDfMaSgGe" lemma="..."/>
          <t id="t9" word="..." tag="NoCmMaSgGe" lemma="..."/>
          <t id="t10" word="..." tag="PTERM_P" lemma="..."/>
        </s>
      </p>
    </body>
  </text>
</cesDoc>
```

TODO

3.3.2 POS Tagging and Lemmatization the set of Documents

TODO

3.3.3 Apply Information Retrieval Methods in order to find the Suggestions

TODO

3.3.4 Adding additional features for the optimization of Machine Learning Processs

TODO

3.3.5 Apply Machine Learning

TODO

3.3.5.1 Selecting Train and Test Set

TODO

```
SELECT
        opngv_suggestion.weight,
        opngv_suggestion.category,
        opngv_suggestion.total_words,
        opngv_trainset.Suggestion
FROM
        opngv_sentence
        INNER JOIN opngv_suggestion
                ON opngv_sentence.comment_id = opngv_suggestion.comment_id
                AND opngv_sentence.sentence_id = opngv_suggestion.sentence_id
        INNER JOIN opngv_trainset
                ON opngv_sentence.comment_id = opngv_trainset.comment_id
                AND opngv_sentence.sentence_id = opngv_trainset.sentence_id
ORDER BY
        opngv_trainset.Suggestion DESC
LIMIT 366
```

TODO

3.3.5.2 Machine Learning Process

TODO

3.3.5.3 Machine Learning Algorithms

TODO

3.4 Overall Opinion Extraction

TODO

3.4.1 Translate Documents to English

TODO

3.4.2 Perform Sentiment Analysis

TODO

- **SentiStrength[2]:** *"SentiStrength estimates the strength of positive and negative sentiment in short texts, even for informal language. It has human-level accuracy for short social web texts in English, except political texts. SentiStrength reports two sentiment strengths:*

  - *-1 (not negative) to -5 (extremely negative)*
  - *1 (not positive) to 5 (extremely positive)*

  Why does it use two scores? Because research from psychology has revealed that we process positive and negative sentiment in parallel - hence mixed emotions. SentiStrength can also report binary (positive/negative), trinary (positive/negative/neutral) and single scale (-4 to +4) results."

- **Sentiment Analysis with Python NLTK Text Classification[3]:** *"Sentiment analysis using a NLTK 2.0.4 powered text classification process. It can tell you whether it thinks the text you enter below expresses positive sentiment, negative sentiment, or if it's neutral. Using hierarchical classification, neutrality is determined first, and sentiment polarity is determined second, but only if the text is not neutral."*

[2] `http://sentistrength.wlv.ac.uk`

[3] `http://text-processing.com/docs/sentiment.html`

4

Evaluation and Results

TODO

4.1 Argument Extraction

TODO

4.1.1 Argument Markers

TODO

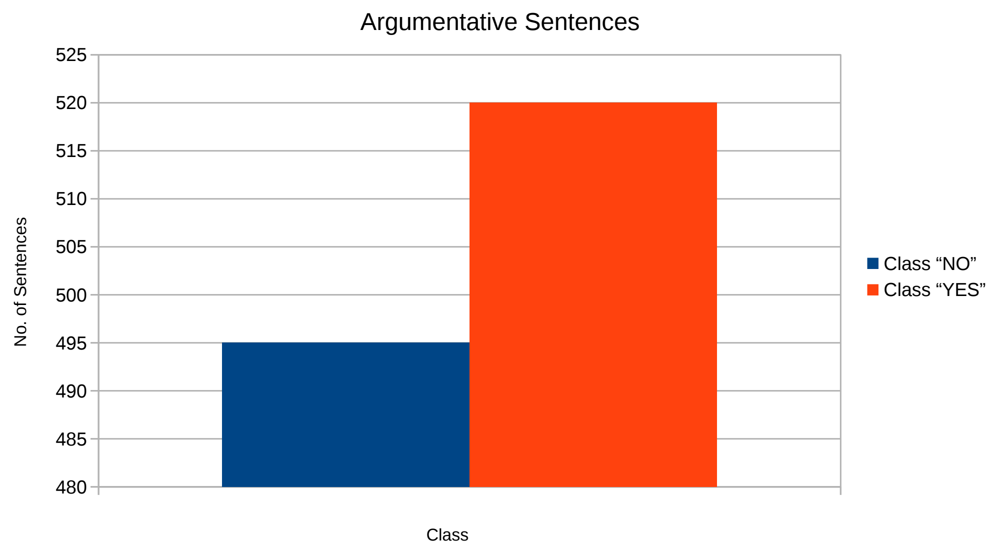

Figure 4.1: Argumentative Sentences in Train Set.

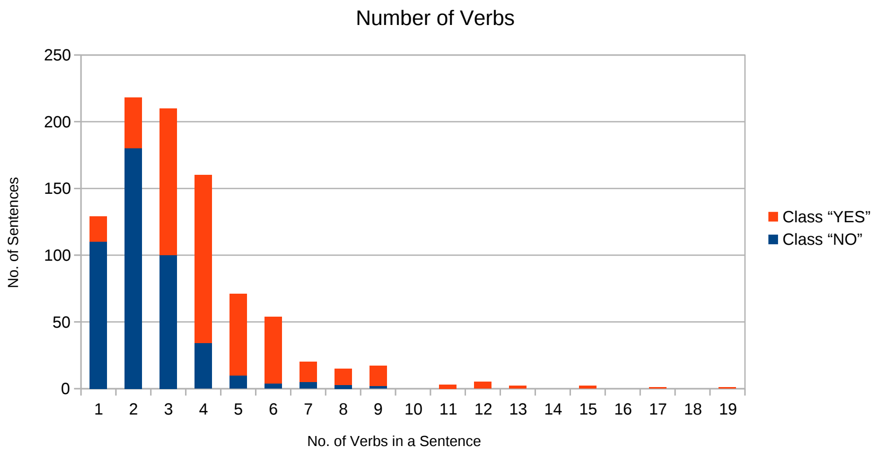

Figure 4.2: Argument Marker - Number of Verbs in a Sentence.

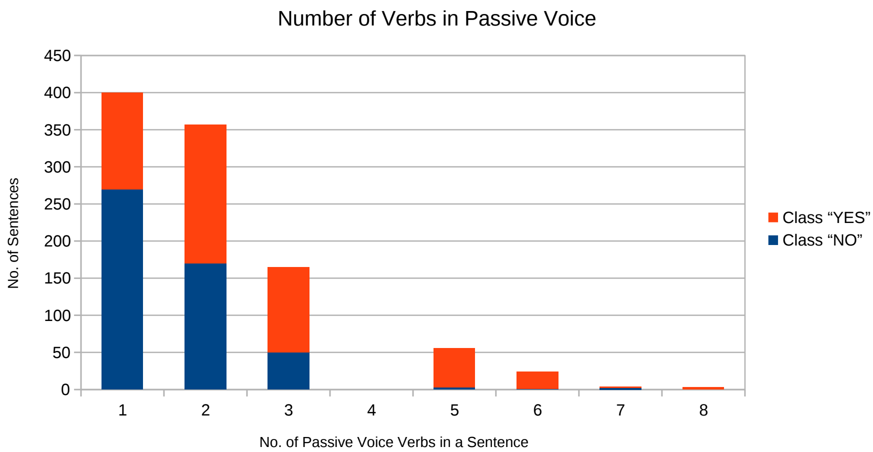

Figure 4.3: Argument Marker - Number of Verbs in Passive Voice in a Sentence.

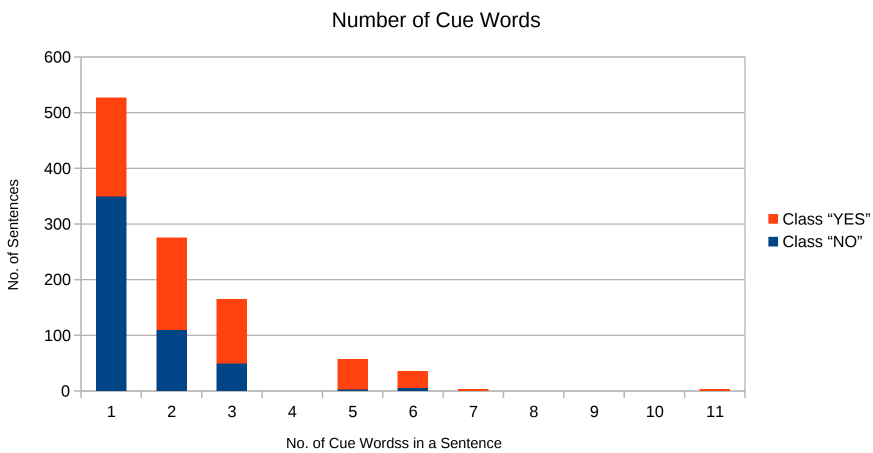

Figure 4.4: Argument Marker - Number of Cue Words in a Sentence.

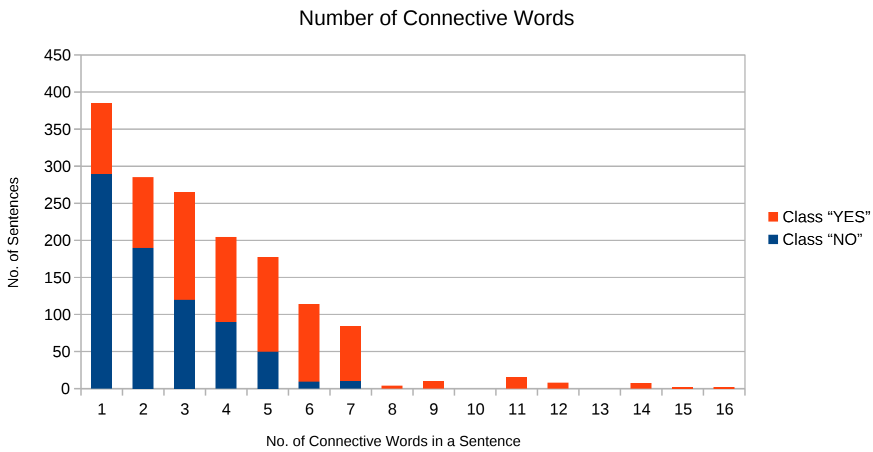

Figure 4.5: Argument Marker - Number of Connective Words in a Sentence.

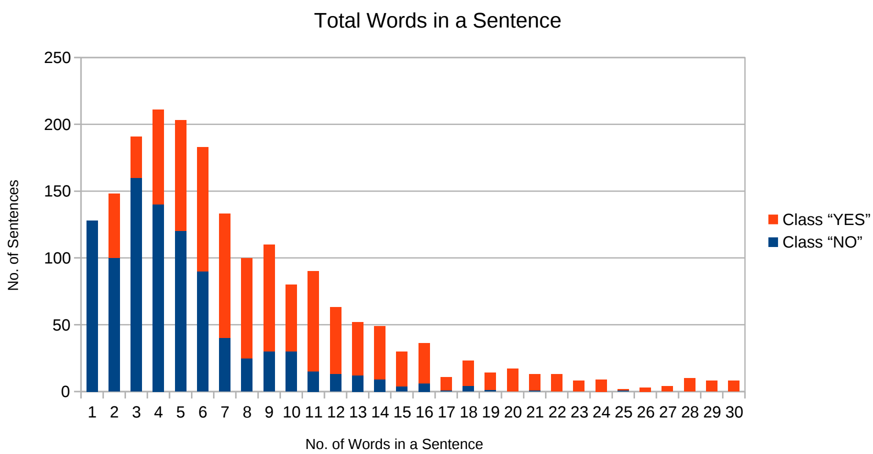

Figure 4.6: Argument Marker - Total words in a Sentence.

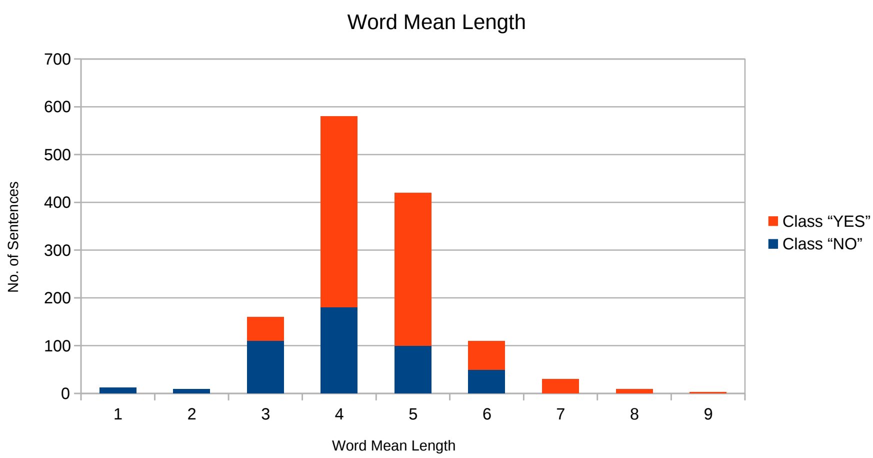

Figure 4.7: Argument Marker - Word Mean Length.

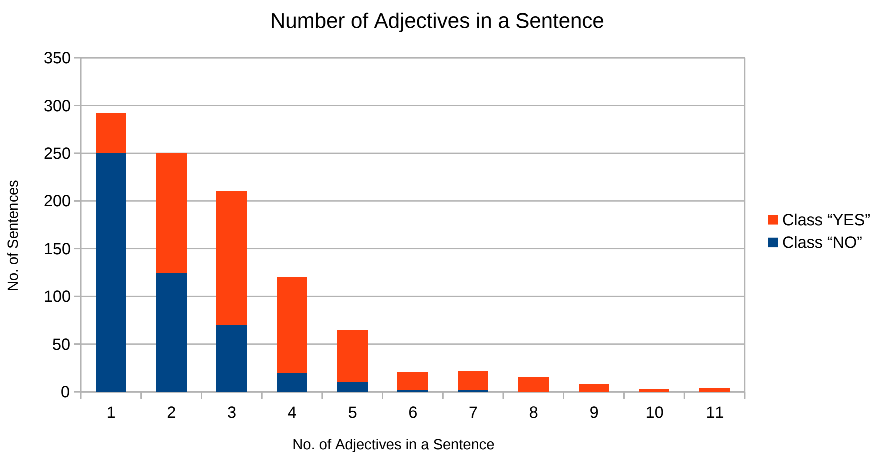

Figure 4.8: Argument Marker - Number of Adjectives in a Sentence.

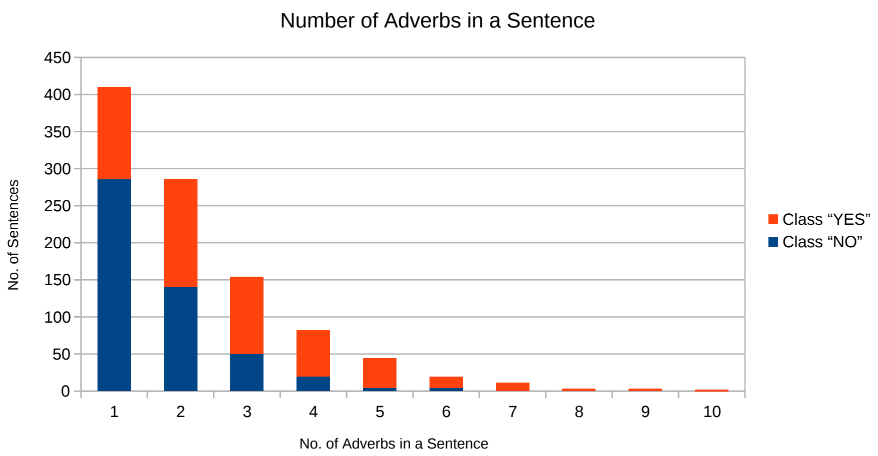

Figure 4.9: Argument Marker - Number of Adverbs in a Sentence.

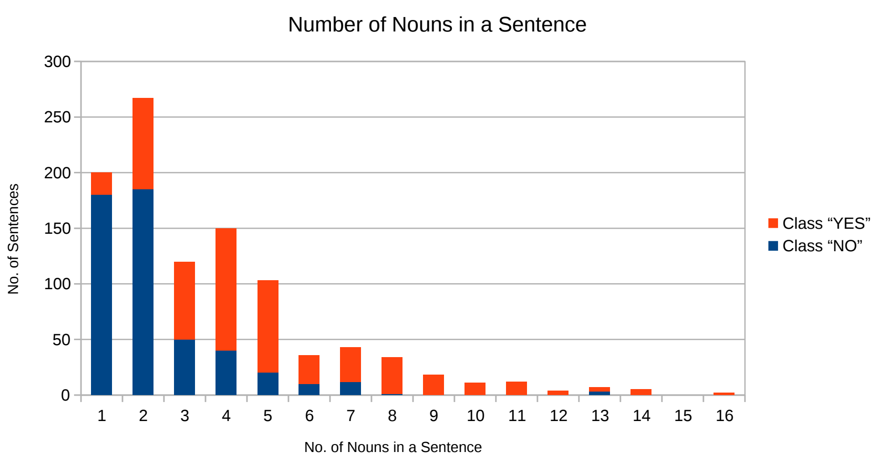

Figure 4.10: Argument Marker - Number of Nouns in a Sentence.

4.1.2 Algorithms used in Machine Learning Procedure

TODO

Table 4.1: Detailed Accuracy for Class "No" (Argument Extraction).

Algorithm	**Precision**	**Recall**	**F-Measure**
SVM	*0.815*	*0.830*	**0.823**
Random Forest	**0.818**	*0.818*	*0.818*
Native Bayes	*0.718*	**0.899**	*0.798*
Logistic Regression	*0.801*	*0.819*	*0.819*

Table 4.2: Detailed Accuracy for Class "Yes" (Argument Extraction).

Algorithm	**Precision**	**Recall**	**F-Measure**
SVM	*0.836*	*0.821*	**0.828**
Random Forest	*0.827*	**0.827**	*0.827*
Native Bayes	**0.873**	*0.663*	*0.754*
Logistic Regression	*0.837*	*0.802*	*0.819*

Table 4.3: Weighted Average on both Classes (Argument Extraction).

Algorithm	**Precision**	**Recall**	**F-Measure**
SVM	**0.826**	**0.826**	**0.826**
Random Forest	*0.823*	*0.823*	*0.823*
Native Bayes	*0.797*	*0.778*	*0.776*
Logistic Regression	*0.820*	*0.819*	*0.819*

TODO

Table 4.4: Additional Statistical Information (Argument Extraction).

	Frequenncy	**Percentage**
Correctly Classified Instances	*838*	*82.56%*
Incorrectly Classified Instances	*177*	*17.4383*
Kappa statistic	*0.6512*	-
Mean absolute error	*0.1744*	-
Root mean squared error	*0.4176*	-
Relative absolute error	-	*34.90%*
Root relative squared error	-	*83.54%*
Coverage of cases (0.95 level)	-	*82.56%*
Mean rel. region size (0.95 level)	-	*50%*
Total Number of Instances	*1015*	-

4.1.3 Information about the Train Set

TODO

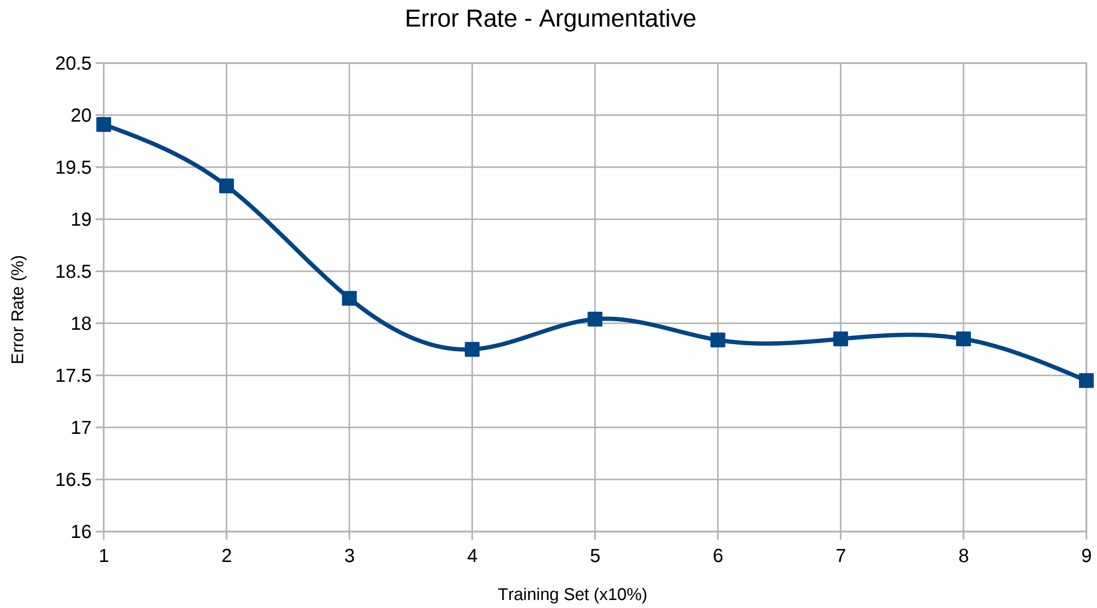

Figure 4.11: Error Rate of Argumentative Sentence Classification.

4.2 Suggestion Extraction

TODO

4.2.1 "10 Fold Cross Validation" on Train Set

TODO

Table 4.5: Detailed Accuracy for Class "No" (Suggestion Extraction).

Algorithm	**Precision**	**Recall**	**F-Measure**
J48	*0.881*	*0.923*	*0.901*
Random Forest	*0.890*	*0.915*	*0.902*
Native Bayes	**0.912**	*0.915*	*0.604*
SVM	*0.839*	**0.989**	**0.908**

Table 4.6: Detailed Accuracy for Class “Yes” (Suggestion Extraction).

Algorithm	Precision	Recall	F-Measure
J48	*0.552*	*0.432*	*0.485*
Random Forest	*0.556*	*0.489*	*0.519*
Native Bayes	*0.608*	**0.601**	**0.604**
SVM	**0.735**	*0.137*	*0.230*

Table 4.7: Weighted Average on both Classes (Suggestion Extraction).

Algorithm	Precision	Recall	F-Measure
J48	*0.822*	*0.834*	*0.826*
Random Forest	*0.830*	*0.837*	*0.833*
Native Bayes	**0.858**	**0.858**	**0.858**
SVM	*0.820*	*0.835*	*0.786*

TODO

Table 4.8: Additional Statistical Information (Suggestion Extraction).

	Frequenncy	Percentage
Correctly Classified Instances	*871*	*85.81%*
Incorrectly Classified Instances	*144*	*14.19%*
Kappa statistic	*0.518*	-
Mean absolute error	*0.1901*	-
Root mean squared error	*0.3382*	-
Relative absolute error	-	*64.21%*
Root relative squared error	-	*87.98%*
Coverage of cases (0.95 level)	-	*95.67%*
Mean rel. region size (0.95 level)	-	*70.64%*
Total Number of Instances	*1015*	-

4.2.2 Equivalent Train Set

TODO

Table 4.9: Detailed Accuracy for Class "No" (Suggestion Extraction, using Equivalent Train Set).

Algorithm	Precision	Recall	F-Measure
J48	*0.940*	*0.810*	*0.870*
Random Forest	**0.996**	*0.810*	**0.893**
Native Bayes	*0.941*	**0.828**	*0.881*
SVM	*0.939*	*0.819*	*0.875*

Table 4.10: Detailed Accuracy for Class "Yes" (Suggestion Extraction, using Equivalent Train Set).

Algorithm	Precision	Recall	F-Measure
J48	*0.470*	*0.765*	*0.582*
Random Forest	**0.533**	**0.984**	**0.641**
Native Bayes	*0.495*	*0.765*	*0.601*
SVM	*0.479*	*0.760*	*0.588*

Table 4.11: Weighted Average on both Classes (Suggestion Extraction, using Equivalent Train Set).

Algorithm	Precision	Recall	F-Measure
J48	*0.855*	*0.802*	*0.818*
Random Forest	**0.912**	**0.841**	**0.857**
Native Bayes	*0.861*	*0.817*	*0.831*
SVM	*0.856*	*0.808*	*0.823*

TODO

Table 4.12: Additional Statistical Information (Suggestion Extraction, using Equivalent Train Set).

	Frequenncy	Percentage
Correctly Classified Instances	*854*	*84.14%*
Incorrectly Classified Instances	*161*	*15.86%*
Kappa statistic	*0.5966*	-
Mean absolute error	*0.2273*	-
Root mean squared error	*0.3467*	-
Relative absolute error	-	*47.98%*
Root relative squared error	-	*73.02%*
Coverage of cases (0.95 level)	-	*97.14%*
Mean rel. region size (0.95 level)	-	*83.00%*
Total Number of Instances	*1015*	-

4.3 Overall Opinion Extraction

TODO

5

Demo Application

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

6
Conclusion

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Bibliography

[1] Frisk, D. (2015) A Chalmers University of Technology Master's thesis template for LaTeX. Unpublished.