

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

MSc Thesis

Speech quality and sentiment analysis on the
Hellenic Parliament proceedings

Konstantina Dritsa

supervised by

Assoc. Prof. Panos Louridas

February, 2018



Table of Contents

1. Introduction	5
2. Sentiment Analysis	7
2.1 Theoretical background	7
2.2 Related Work	8
3. Speech Quality & Readability Formulas	11
3.1 Theoretical Background	11
3.2 Related Work	13
3.2.1 Flesch Reading Ease	13
3.2.2 FOG index	14
3.2.3 SMOG Index	14
3.2.4 Limitations	15
3.2.5 Applications in the Greek Language	15
4. Methodology	16
4.1 Data Collection	16
4.1.1 Parliament Records Retrieval	16
4.1.2 Parliament Members Information Retrieval	20
4.2 Data Preprocessing	25
4.2.1 Members Data Cleaning	25
4.2.2 Records Cleaning & Data Extraction	27
4.3 Analysis	39
4.3.1 Introduction	39
4.3.2 SMOG Index Calculator for the Greek Language	40
4.3.3 Greek Lexicons for Sentiment Analysis	43
Lexicon 1: Greek Sentiment Lexicon with 6 Sentiment Scores	43
Lexicon 2: Greek Sentiment Lexicon with Positive & Negative Sentiment scores	44
4.3.4 Word Identification	45
4.3.5 Mathematical formula of Sentiment Evaluation	47
4.3.6 Mathematical formula of Readability Evaluation	49
4.3.7 Summary of Analysis Implementation	49
5. Results	53
5.1 Speech quality results	53
5.2 Sentiment analysis results based on Lexicon 1	56
5.3 Sentiment analysis results based on Lexicon 2	61
6. Conclusion	66



7. Future work	67
8. References	68



Abstract

“It's not what you say, but how you say it”. How often have you heard that phrase? Have you ever wished that you could take an objective and comprehensive look into what is said and how it is said in politics? Within this project, we examined the records of the Hellenic Parliament sittings from 1989 up to 2017 in order to evaluate the speech quality and examine the palette of sentiments that characterize the communication among its members. The readability of the speeches is evaluated with the use of the “Simple Measure of Gobbledygook” (SMOG) formula, partially adjusted to the Greek language. The sentiment mining is achieved with the use of two Greek sentiment lexicons. Our findings indicate a significant drop on the average readability score of the parliament records from 2003 up to 2017. On the other hand, the sentiment analysis presents steady scores throughout the years. The communication among parliament members is characterized mainly by the feeling of surprise followed closely by anger and disgust. At the same time our results show a steady prevalence of positive words over negative. The results are presented in graphs, mainly in comparison between political parties as well as between time intervals.



Acknowledgments

I would like to express my gratitude to my esteemed supervisor, Prof. Panos Louridas, for his undivided guidance and crucial insight throughout this project. His knowledgeability, way of thinking and passion for discovery have fueled me with excitement, inspiration and patience.

I would also like to thank my family, friends and colleagues for their care and support and especially Dimitris for his encouragement and the helpful discussions we had.



1. Introduction

The onslaught of information that technology has brought to our lives often makes us feel powerless and lost. And it can get worse when politics are the subject of interest. But, at the same time, technology can provide a helping hand to put information into perspective and have an objective outlook on the status quo as well as the past of politics.

In this project, as writings remain, we examined 4905 record files of the Hellenic Parliament sittings from 1989 up to 2017, which we gathered by crawling the website of the Hellenic Parliament. The initial size of the 4905 downloaded records was 3.18 GB. The number of unique parliament members that we examined from 1989 and onwards was 1491. The speeches we distinguished from the records and matched with their corresponding speakers were 999,399. These were grouped by speaker per sitting in 115,241 speeches.

We, then, evaluated our dataset in regard to the readability of the speeches and the sentiments that characterize the interactions among the parliament members.

For the readability evaluation of the speeches we utilized the SMOG Index formula. The application of the SMOG Index on the Greek language was a challenge, as it consists of a topic not thoroughly examined, without ready-made tools for its implementation. Our findings showed a generalized decrease in the SMOG Index since the early 2000s. The downward trend is stronger for the political parties DIMAR and Golden Dawn, which show the lowest SMOG Index in the recent years during which they were elected.

Concerning the sentiment analysis, we used two different Greek sentiment lexicons and evaluated the presence of a range of different sentiments. The sentiment ratings of our dataset are a result of direct calculations derived from the words constructing the speeches. Our results show that the speeches and discussions that take place in the Hellenic Parliament sittings are significantly characterized by the sentiment of surprise, followed closely by those of anger and disgust. The presence score of these sentiments lies around 3.5 in a scale from 0 to 5. Happiness and fear are represented with a score around 2. The sentiment score of sadness is the lowest with a value around 1. Furthermore, the percentage of positively charged words is slightly but steadily prevailing over the percentage of negatively charged words, especially in the last decade.



The results are presented in graphs, mainly in comparison between political parties as well as between time intervals.

The remainder of the thesis is organized as follows. In section 2, we make an introduction to the field of sentiment analysis, its theoretical bases and common research practices. In section 3, we dive into the basics of the text readability evaluation, its theoretical background, the related work and the common formulas created for the measurement of the readability and speech quality of a text. We also refer to the application of readability formulas on the Greek language and its limitations. In the lengthy section 4, we describe our full methodology. This section includes all the steps for the collection of our data from the website of the Hellenic Parliament (section 4.1), the preprocessing of the data in order to clean unwanted information and format them in a way that can facilitate the extraction of our desired results (section 4.2), the preparation of our tools for the analysis implementation (section 4.3) and last but not least the detailed description of the analysis implementation (section 4.4). Finally, section 5 includes a presentation of our results with graphic animation, section 6 sums up the steps of the project and its conclusions and section 7 suggests some future work.

Our contributions are:

- The extraction of data from the records of the Hellenic Parliament sittings and the creation of a dataset that can be used in the future for extracting further interesting findings.
- The development of a customized and effective process to match each speech in the Hellenic Parliament sitting records with the name of the official parliament member that gave the speech.
- The application of SMOG Index on the Greek language and the identification of specific deficiencies in the published literature that create room for further research on this topic.
- An objective overview on the readability and the sentiments that characterize the speeches and communication among the members of the Hellenic Parliament from 1989 until 2017.



- The development of an efficient identification process of the corpus words with the words of the sentiment lexicons we used, customized to the specialized vocabulary of the Hellenic parliament.

2. Sentiment Analysis

2.1 Theoretical background

Sentiment analysis is an interdisciplinary field that crosses natural language processing, artificial intelligence, and text mining. As most opinions are available in text, a format more easily processable than others, sentiment analysis is assumed to have emerged as a subfield of text mining [9]. Sentiment Analysis is the computational study of people's opinions, attitudes and emotions toward an entity. The entity can represent individuals, events or topics [7]. It generally analyzes opinions of people expressed in text, aiming to classify its polarity as positive or negative. Due to its many aspects it is often referred to with different names such as opinion mining, sentiment classification, subjectivity analysis, and sentiment extraction [8]. However, some researchers stated that there are slight differences among these notions [6].

Sentiment analysis appeared in the literature in the 1990s. The rapid growth of the user-generated content represented in social media platforms, blogs, wikis and web forums have made it an increasingly popular research topic in Information Retrieval and web data analysis, especially after 2000 [8].

Sentiment analysis and opinion aggregation on large scale data can have a great impact and can provide a helpful insight in many different domains such as shopping, entertainment, politics, education, and marketing [9-12]. The targets of sentiment analysis usually include products, services, topics, and social issues [11-16].

As sentiment analysis can be approached as a classification process, we can note three main classification levels:



- Document-level sentiment analysis: It considers the whole document a basic information unit talking about a topic and aims to classify it as having positive or negative sentiment towards the topic.
- Sentence-level sentiment analysis: It considers each sentence as a basic information unit. Firstly, it identifies the sentence as subjective or objective. Then, if the sentence is subjective, it aims to classify it as having positive or negative sentiment.
- Aspect-level sentiment analysis: It aims to classify the sentiment with respect to the specific aspects of entities. This presupposes the identification of the entities and their aspects. An example of different opinions for different aspects of the same entity could look like this “The thesis related work is not thoroughly examined, but the thesis results are very interesting” [6].

It is important to note here that there is no fundamental difference between document and sentence level classifications, as sentences can be seen as short documents [10].

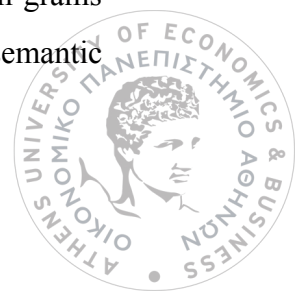
2.2 Related Work

Much of the existing research in sentiment analysis elaborates on two major topics: the features utilized to represent the unit of text [19] and the techniques chosen for the implementation of sentiment analysis [17].

Sentiment Analysis Features

Several classes of features have been applied in the literature, of which the following four categories stand out.

- Semantic features: These mainly include manually or semi-automatically generated sentiment lexicons of specific terms labeled as expressing positive or negative sentiment [13,19]. Other semantic attributes include contextual features representing the semantic orientation of surrounding text, which have been useful for sentence level sentiment classification [21].
- Syntactic features: These mainly include word n-grams and part-of-speech n-grams (for example frequency of part-of-speech tags, [12, 17, 20]). Along with semantic



features, syntactic attributes are the most commonly used set of features for sentiment analysis [19].

- Stylistic features: These mainly include lexical and structural attributes. For example the frequency of letters (e.g., a, b, c) or the occurrence of special characters (#\$%^&*). However, lexical and structural style markers have seen limited usage in sentiment analysis research. [19]
- Link-based features: These mainly refer to link/citation analysis for the evaluation of sentiments for web resources and documents. It has been shown that opinion web pages heavily linking to each other often share similar sentiments [22]. However, due to the limited usage of link-based features, it is unclear how effective they may be for sentiment classification [19].

Due to the selection of a large amount of features, researchers often resort to feature selection prior to classifier calibration, to identify and use the most significant discriminators of opinion expression for each case [5,19,20].

Sentiment Analysis Techniques

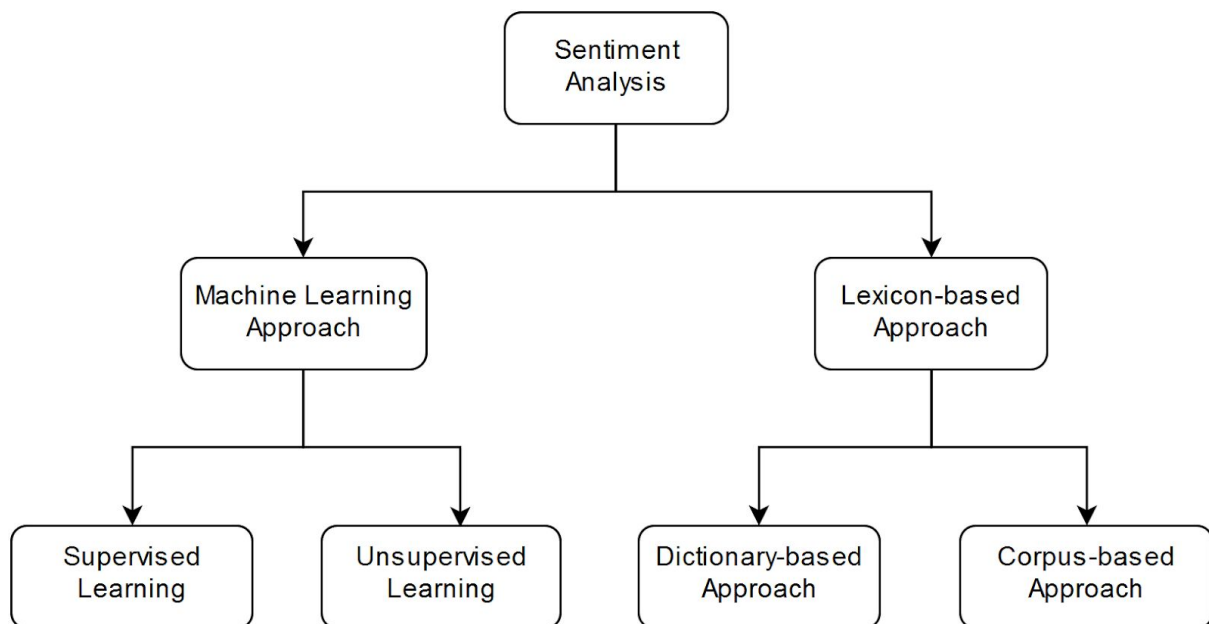


Figure 1: Sentiment analysis techniques

Techniques for sentiment analysis can be broadly categorized into two main classes.

- *Machine Learning Approaches*

The first class of approaches utilize a textual feature representation mixed with machine learning algorithms to derive the relationship between features of the text segment and the opinions expressed in the writing, in either a supervised or an unsupervised way [5,17]. With the term supervised, we mean that the classifier receives a set of manually labeled examples as training data and makes predictions for all unseen points [18]. With the term unsupervised, we mean that the classification receives unlabeled data and makes evaluations or predictions for all unseen points [18]. Labels can be assigned to training instances manually through human evaluation of the text, or can be predefined with ratings, such as the number of stars in a review [5]. Semi-supervised and unsupervised techniques are proposed when there is no initial set of labeled data for training the classification algorithm [44, 45]. Furthermore, hybrid approaches, combining supervised and unsupervised techniques, or even semi-supervised techniques, can be used to classify sentiments [46, 47]. Concerning the feature selection for the detection of sentiment, Natural Language Processing plays an important role, as it provides useful commonly used features such as the frequency of terms, part of speech information and syntactic dependencies [4].

Many prominent approaches to sentiment analysis utilize machine learning techniques to develop classifiers [12,17,19,20]. Among the various machine learning techniques, Support-Vector Machines (SVMs) and Naïve Bayes are most commonly applied [4,5,17]. The use of deep learning, artificial neural networks (such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks) and maximum entropy models have also demonstrated success in sentiment analysis applications, gaining more and more popularity [5]. Support Vector Machines are also combined with neural network methods or with dense word embedding features [1,5].



- *Lexicon-Based Approaches*

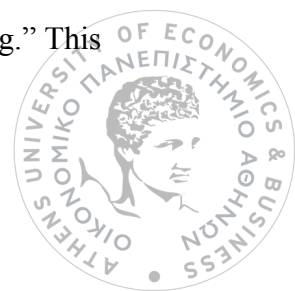
The second class bases the evaluation of a text on a sentiment lexicon of opinion-related and usually manually defined positive or negative terms in an unsupervised way [4,5,13]. In this class, one can find two subcategories. The dictionary based approach, where the analysis is based on the use of an initial set of terms that are usually manually collected and annotated and the corpus-based approach, where the analysis is based on domain-related dictionaries. These dictionaries are generated from a set of terms that grows through the search of related words with statistical or semantic techniques [4].

A dictionary-based sentiment analysis methodology for the Greek language that significantly influenced our work is the one implemented in the paper “Sentiment Analysis of Greek Tweets and Hashtags using Sentiment Lexicon” (Mallis, Kalamatianos, Nikolaras, Symeonidis, 2014) [2]. In that paper, a Greek sentiment lexicon was used [36] that included a list of Greek words and an arithmetic evaluation of 6 sentiments for each word, by 4 annotators. The results of the paper included a 6-sentiment vector representation of each tweet and consequently of each hashtag examined. Each sentiment in the tweet vector was calculated by the quadratic mean of the sentiments of the lexicon words identified in the tweet. Accordingly, each sentiment in the hashtag vector was created by the quadratic mean for the sentiments of the tweets it was mentioned in. From this paper, we used the same Greek sentiment lexicon and the same mathematical equations for sentiment analysis.

3. Speech Quality & Readability Formulas

3.1 Theoretical Background

Readability, as introduced in the context of readability formulas, is defined as the ease with which a reader can understand a written text. It is often confused with legibility, which concerns the presentation, such as font size and line length. George Klare [26] defines readability as “the ease of understanding or comprehension due to the style of writing.” This



definition focuses on the content as separate from presentation issues such as layout and organization. The creator of the SMOG readability formula G. Harry McLaughlin [27] defines readability as “the degree to which a given class of people find certain reading matter compelling and comprehensible.” Edgar Dale and Jeanne Chall [28] comprehensively identify readability as “the sum total (including all the interactions) of all those elements within a given piece of printed material that affect the success a group of readers have with it. The success is the extent to which they understand it, read it at an optimal speed, and find it interesting.”

The significance of readability measurement and formulas is highlighted when theory meets reality in very important issues. In 1998, in the United States, traffic accidents caused 46 percent of all accidental deaths of infants and children aged 1 to 14, with the single strongest risk factor for injury in a traffic accident being the improper use of child-safety seats [23]. A research was conducted on the installation instructions of the child seats with the use of the SMOG readability formula. The authors concluded that the average reading level of the instructions corresponded to a reader’s knowledge level of the 10th grade. This was too difficult for 80 percent adult readers in the United States [24].

In general, the main factors that affect the readability level of a text are the content, meaning the complexity of its vocabulary and syntax, and the legibility, such as font size and line length. It has been recognised that there is a range of reader factors which affect the reading and comprehensibility process. Some variables are beyond the control of the information transmitter, such as motivation and reading experience. But there are also variables that can be adjusted to the readers’ abilities, that is facets of the text. Among those, most often used to assess text difficulty are sentence length and vocabulary difficulty. In the 1920s, educators discovered a way to use these two factors in order to predict the difficulty level of a text. They embedded this method in readability formulas, which have proven their worth in over 80 years of application [25].



3.2 Related Work

There is a list of readability measures that have been invented and tested through the years, among which the most popular are the following:

3.2.1 Flesch Reading Ease

The one perhaps most responsible for spreading the importance of readability was Rudolf Flesch. In his dissertation, Flesch published his first readability formula for measuring adult reading material, that could increase readership by 40 to 60 percent.

In 1948, he published a second formula with two parts. The first part, the Reading Ease formula, used only two variables, the number of syllables and the number of sentences for each 100-word sample. The reading ease is measured within a scale from 0 (difficult) to 100 (easy). The second part of Flesch's formula predicts human interest by counting the number of personal words (such as pronouns and names) and personal sentences (such as quotes, exclamations, and incomplete sentences).

The formula for the updated Flesch Reading Ease score is:

$$206.835 - 1.015 * \left(\frac{\text{total words}}{\text{total sentences}} \right) - 84.6 * \left(\frac{\text{total syllables}}{\text{total words}} \right)$$

Flesch's Reading Ease became one of the most popular, tested and reliable readability formulas [26]. In 1976, a study commissioned by the U.S. Navy modified the Reading Ease formula to produce a grade-level score. This popular formula is known as the Flesch-Kincaid formula, the Flesch Grade-Scale formula or the Kincaid formula [25].

$$0.39 * \left(\frac{\text{total words}}{\text{total sentences}} \right) + 11.8 * \left(\frac{\text{total syllables}}{\text{total words}} \right) - 15.59$$

The result is a number that corresponds with a U.S. grade level.



3.2.2 FOG index

Robert Gunning in “The Technique of Clear Writing” [31] published his readability formula “Fog Index”, deriving from the concept of “fog” and unnecessary complexity he noticed in the newspapers and business texts. The Fig index requires a word sample of at least around 100 words. The complete formula is:

$$0.4 * \left[\left(\frac{\text{words}}{\text{sentences}} \right) + 100 * \left(\frac{\text{complex words}}{\text{words}} \right) \right]$$

Where complex words = number of words of more than two syllables, while excluding from this list proper nouns, familiar jargon, or compound words. Common suffixes (such as -es, -ed, or -ing) should not be counted as a syllable. Concerning the results of the index, 5 means the text is easy to read while 20 means it is very difficult. A commonly used guideline is that the average 15-year-old can cope with texts with an index of 10 while the average level for university students is 14-16 [32]. The Fog Index became popular because of its ease of use [25].

3.2.3 SMOG Index

G. Harry McLaughlin published his SMOG formula in the belief that the word length and sentence length should be multiplied rather than added. By counting the number of words of more than two syllables in a sample text of 30 sentences, he provides this simple formula:

$$\text{grade} = 1.0430 * \sqrt{\text{number of polysyllables} * \frac{30}{\text{number of sentences}}} + 3.1291$$

The SMOG Index gives as output the U.S. grade level that a person must have reached in order to fully understand a text. This simply means that the higher the SMOG Index, the more difficult a text is to be understood.



Furthermore, measurements for texts of fewer than 30 sentences are statistically invalid, because the formula was normed on 30-sentence samples. For texts with more than 30 sentences, the sample text of the 30 sentences will have to consist of 10 consecutive sentences near the beginning, 10 from the middle and 10 near the end of the initial text. As sentence is counted any string of words ending with a period, question mark or exclamation point. It is also stated that “any string of letters or numerals beginning and ending with a space or punctuation mark should be counted if you can distinguish at least three syllables when you read it aloud in context”. According to G. H. McLaughlin, the linguistic measures which have been found to have greatest predictive power are word and sentence length [25,27].

3.2.4 Limitations

Readability formulas have been created as an easy and fast means of evaluating the reading difficulty of a text. However, they have been a subject of thorough critique concerning the correctness of their criteria. For example, limiting the factors under consideration to the word and sentence length, leaves out the importance of proper grammar or meaning of content [34]. Furthermore, another common argument was that not all long words are hard and complex and not all short words are easy [25]. We should underline here that reading formulas were never proposed as the sole medium to evaluate the reading difficulty of a text. They work more like an indicator that can provide warnings.

3.2.5 Applications in the Greek Language

Most of the reading formulas, including the aforementioned indexes, were tested and validated in English texts, thus creating a probability of bias when applied in other languages. It is likely that natural variations among languages regarding the various predictors of reading difficulty (length of words, length of sentences) could lead to different regression equations [33]. Especially concerning the modern Greek language, the basic difference with the English language is that Greek words are on average longer, meaning that they have more syllables.



In the beginning of 1980s, Gagatsis was the first to deal with the readability of Greek texts, studying the readability of the Mathematics school manuals. He adjusted the Flesch Reading Ease formula to the Greek language, based on the observation that Greek words are on average longer than the English or the French words. So he replaced the number “84.6” of the formula that represented the average number of syllables per word with the number “59”. The new formula was remodeled as follows [35]:

$$R = 206.835 - 1.015 * \left(\frac{\text{total words}}{\text{total sentences}} \right) - 59 * \left(\frac{\text{total syllables}}{\text{total words}} \right)$$

At the beginning of the first decade of the 21st century, a new software for evaluation of the readability of Greek texts was created by the Center of the Greek Language, a research institute in Thessaloniki. The “grval 1.1” (from Greek evaluation) was based in adaptations in Modern Greek of the Flesch Reading Ease (average sentence length, average syllables per word), Flesch-Kincaid Grade Level, Flesch Fog Index (prefixes, suffixes) and SMOG (polysyllabic words). Unfortunately, we could not find an online working interface of the software or information on how all these formulas were adjusted to the Greek language. We also contacted the people responsible for the project, but they could not provide us with any information.

4. Methodology

4.1 Data Collection

4.1.1 Parliament Records Retrieval

For the retrieval of the parliament records and due to the absence of an API, we created a web crawler named “web_crawler.py” for crawling the Hellenic Parliament website, that displays an online catalogue with all the parliament records and their information from 1989 up to today, as you can see in this page <https://www.hellenicparliament.gr/Praktika/Synedriaseis-Olomeleias>. This catalogue is



actually an html table, as shown in the figure below, that includes the date, the parliamentary period, session and sitting that the records belongs to. It also includes links to relative videos (if available) and links to the actual record files in the formats of pdf, doc, docx and/or text. For parsing the html of the web pages, we used the Python library BeautifulSoup4 (<http://beautiful-soup-4.readthedocs.io/en/latest/>).


Βρέθηκαν 5002 συνεδριάσεις Σελίδα 215 από 501					
Ημερομηνία	Περίοδος	Σύννοδος	Συνεδρίαση	Σχετικά Videos	
27/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΠΖ'		 
26/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΠΣΤ'		 
23/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΠΕ'		 
22/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΠΔ'		 
21/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΠΓ'		 
20/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΠΒ'		 
15/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΠΑ'		 
14/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	Π'		 
13/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΟΘ'		
12/02/2007	ΙΑ' ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΚΟΙΝΟΒΟΥΛΕΥΤΙΚΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)	Γ'	ΟΗ'		 
Εγγραφές: 2141 - 2150 από 5002 - Σελίδες:   213 214 215 216 217  					

Figure 2: Screenshot of the Hellenic Parliament website listing of the sitting records

The steps we followed for downloading the parliament record files for each row in each page are showcased with the following figure and further explained below:

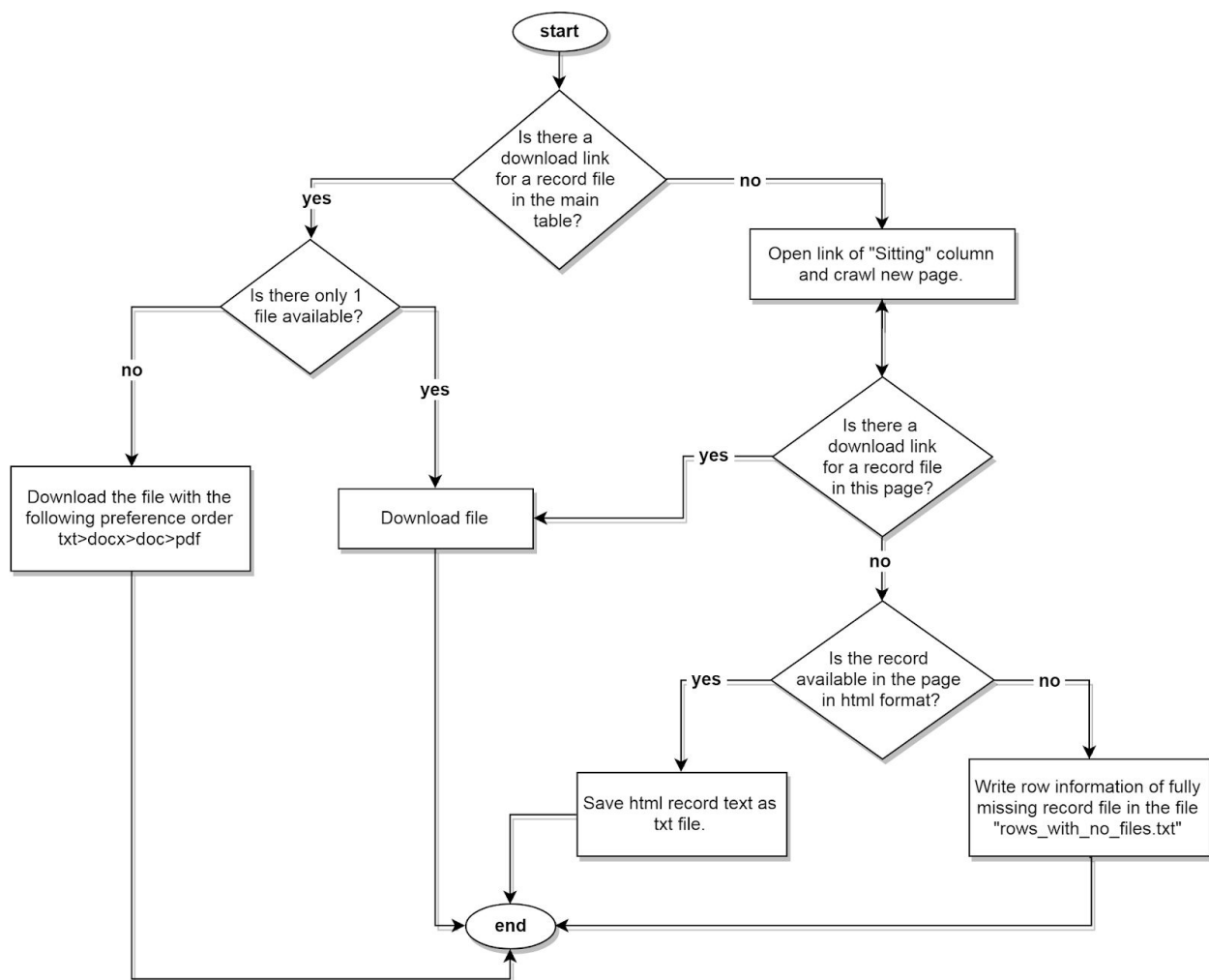


Figure 3: Steps for downloading the Hellenic Parliament sitting records

In some cases, links to the record files are not available in the main table view. They are hidden in the pages one can visit by clicking on the HTML links connected with the "Sitting" column entries in the main table. These pages can display a link to the record file in PDF, doc, docx and/or text format or the actual content of the record in the page in HTML format. Sometimes they also display an empty page and the record is missing from the website. The names of the record files did not follow a specific helpful pattern and their pattern actually changed multiple times from 1989 until today, thus not providing useful information.

In order to address these difficulties, we parsed the HTML of each page and we firstly looked for a link to a record file in the main table. Luckily, all these links pointing to record files in the main table have in common the string "/UserFiles/" as part of their URL, so this

constituted our criterion. In case there were more than one links to files with different file formats available for the same record, we would give priority to text files, followed by docx, then doc and lastly PDF files. We chose this order based on which file format was more easily converted to text, especially without any encoding problems. For downloading the files (actually getting the content of the files) we used the Requests library (<http://docs.python-requests.org/en/master/>) and specifically the requests.get() method. In the cases where no link to a record file was available in the main table, we would open the link of the "Sitting" column entry and search for a file in the new page. If there was no file in the new page but the record was available in HTML format throughout the page, we would use the BeautifulSoup4 python library to parse the html, clean up the html tags and save the plain text as a text file. In order to give to the files meaningful names that could facilitate our future work and save all the information available from the main table, we saved each file with the following name pattern:

year-month-day_ascendingCounter_period_session_sitting.fileExtension.

The strings of period, session and sitting were available in Greek language. In the following section “4.2.2 Records Cleaning & Data Extraction” we explain the process of renaming the files to English and converting them all to a text file format.

In a few cases where this information was for some reason missing from the main table, we corrected the names manually. When no record was available for a specific table entry in text, doc, docx, pdf or HTML format, as mentioned above, we wrote the date of the table entry and the page number in a file named "rows_with_no_files.txt". This file finally included details for 12 completely missing sitting record files.



4.1.2 Parliament Members Information Retrieval

The website of the Hellenic Parliament displays a comprehensive list of all the 1787 members of the Hellenic Parliament from the “Restoration of Democracy” (Greek polity change) in 1974 up to 2017.

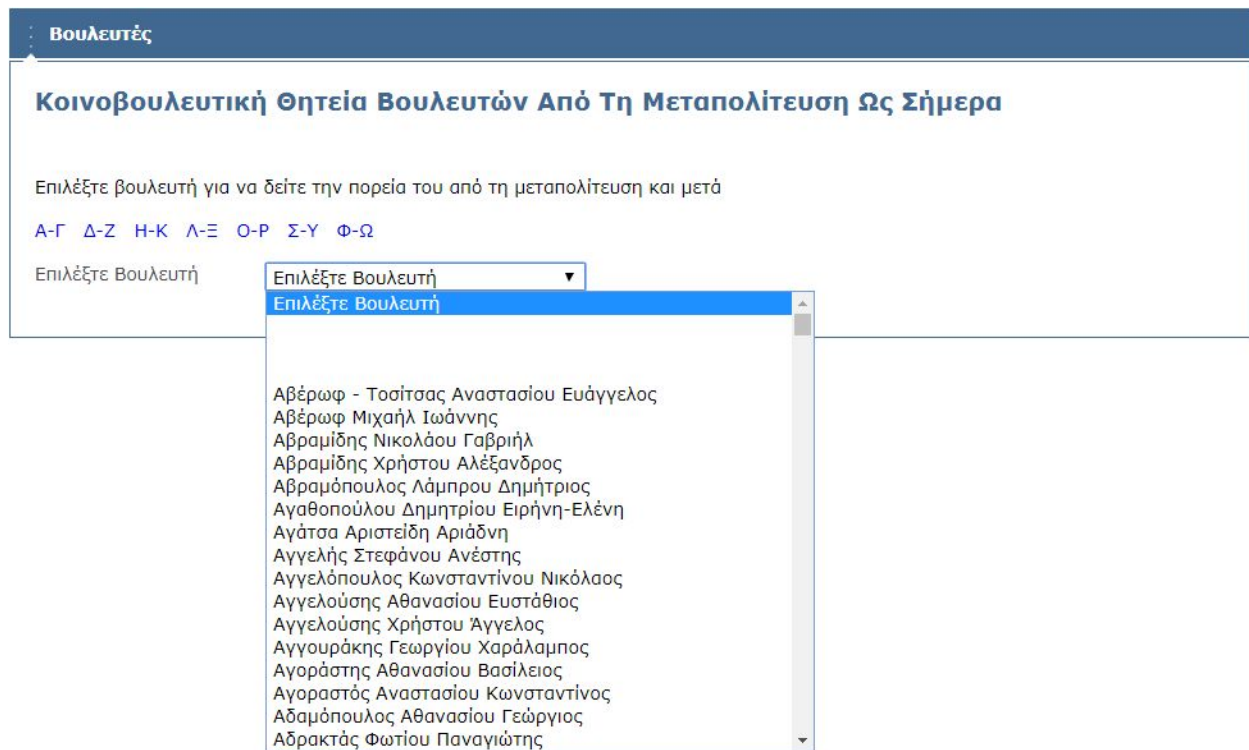


Figure 4: Screenshot of the Hellenic Parliament website listing of its members from 1974 until today

By choosing a member from the list, the website displays an HTML table with information that includes the periods that the member served and for each period the geographical region, political party, a description and a date referring to the description.

Due to the absence of an API for the retrieval of the members' information, we created a web crawler named "web_crawler_parliament_members.py".

To begin with, we obtained the content of the starting page that includes the drop down list

<http://www.hellenicparliament.gr/Vouleftes/Diatelesantes-Vouleftes-Apo-Ti-Metapolitefsi-Os-Simera/>



and we parsed the corresponding HTML elements with the help of the Python library BeautifulSoup4. In particular, in the main page we located the list with option tags, each one representing a parliament member, as you can see in the following figure.

```
<option value=""> Επιλέξτε Βουλευτή</option>
<option value="49ca7f02-b50e-4256-b7fa-a43401404484"></option>
<option value="9afb9452-eef4-4886-ad81-a43401422454"></option>
<option value="d58c4c67-86d5-4d14-a37d-a43400d64a50"></option>
<option value="a64aa47f-1892-484a-9093-9a74ebbded40">Αβέρωφ - Τοσίτσας Αναστασίου Ευάγγελος</option>
<option value="a6851439-c82a-4e4a-a271-d03cbe4216c4">Αβέρωφ Μιχαήλ Ιωάννης</option>
<option value="6d6d25a4-53ab-40d2-84e0-830d3c6fca77">Αβραμίδης Νικολάου Γαβριήλ</option>
<option value="5b5223da-da6c-40e2-9e34-c5a11d28950f">Αβραμίδης Χρήστου Αλέξανδρος</option>
<option value="e7de51bc-72ac-49dd-a848-7b49721dc247">Αβραμόπουλος Λάμπρου Δημήτριος</option>
<option value="013bab92-bddf-4508-b849-6150356536c4">Αγαθοπούλου Δημητρίου Ειρήνη-Ελένη</option>
<option value="4d3c7c34-7ad1-42e2-a807-01ee80db406b">Αγάτσα Αριστεΐδη Αριάδνη</option>
<option value="919a21f5-c973-4c78-9570-07e9a51a815e">Αγγελής Στεφάνου Ανέστης</option>
```

Figure 5: Screenshot of the html select list of all the Hellenic Parliament members from 1974 until today

Some of the options in the list were not valid parliament members. In the image above, such is the case for the first four options. So, we obtained from the select list the name of each valid member along with the value attribute, which is a form of ID. This ID, when added to the website link, opens a new page with information about the parliamentary activity of the member.

https://www.hellenicparliament.gr/Vouleftes/Diatelesantes-Vouleftes-Apo-Ti-Metapolitefsi-Os-Simeraz?MpId=a64aa47f-1892-484a-9093-9a74ebbded40

Επιλέξτε Βουλευτή

Περίοδος	Ημ/νία	Περιφέρεια	Κοιν. ομάδα	Περιγραφή
A' (17/11/1974 - 22/10/1977)	17/11/1974	ΙΩΑΝΝΙΝΩΝ	ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ	Εκλογής
B' (20/11/1977 - 19/09/1981)	20/11/1977	ΙΩΑΝΝΙΝΩΝ	ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ	Εκλογής
Γ' (18/10/1981 - 07/05/1985)	18/10/1981	ΙΩΑΝΝΙΝΩΝ	ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ	Εκλογής
Δ' (02/06/1985 - 02/06/1989)	02/06/1985	ΕΠΙΚΡΑΤΕΙΑΣ	ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ	Εκλογής
Ε' (18/06/1989 - 12/10/1989)	18/06/1989	ΕΠΙΚΡΑΤΕΙΑΣ	ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ	Εκλογής
ΣΤ' (05/11/1989 - 12/03/1990)	05/11/1989	ΕΠΙΚΡΑΤΕΙΑΣ	ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ	Εκλογής
ΣΤ' (05/11/1989 - 12/03/1990)	02/01/1990	ΕΠΙΚΡΑΤΕΙΑΣ	ΝΕΑ ΔΗΜΟΚΡΑΤΙΑ	Απεβίωσε (αντικαταστάτης: Κοραχάης Βασίλειος Ιωάννη)

Figure 6: Screenshot of the information page of a parliament member on the Hellenic Parliament website

This information includes, as mentioned before, the parliamentary periods that this person served as a member, the respective geographical region of Greece for which he/she was elected, the political party that he/she belonged to, a small description and a date field that seems to refer to the content of the small description. We saved the names and link IDs in a Python dictionary, by skipping the first four cases where the name is “Επιλέξτε Βουλευτή” (in English “Choose Parliament Member”) or an empty string.

Then for each entry of the Python dictionary, we recreated the full link of the member page and we obtained the HTML content of the member’s page with the use of the Requests library and the BeautifulSoup4 library. From each such page, we located the HTML table with the information and we saved each line in the file “original_members_data.txt” with the form: ascending counter, name, period, date, administrative region, parliamentary party, description. In some cases the period information was in the form “ΙΖ’(20/09/2015-”, which means that the period has not yet been terminated. So, in order to maintain a consistency in our date, we filled in the period string after the dash with the date that we run this script,

using the Datetime python library (<https://docs.python.org/2/library/datetime.html>). Furthermore, we used regular expressions to remove noise from our data such as newlines, whitespace, and tabs.

In the cases where the member page had no table entries and it is in fact an empty page, we wrote in the file the ascending counter, the member name and the note “NO DATA”.

The procedure is probably better shown in the diagram below:



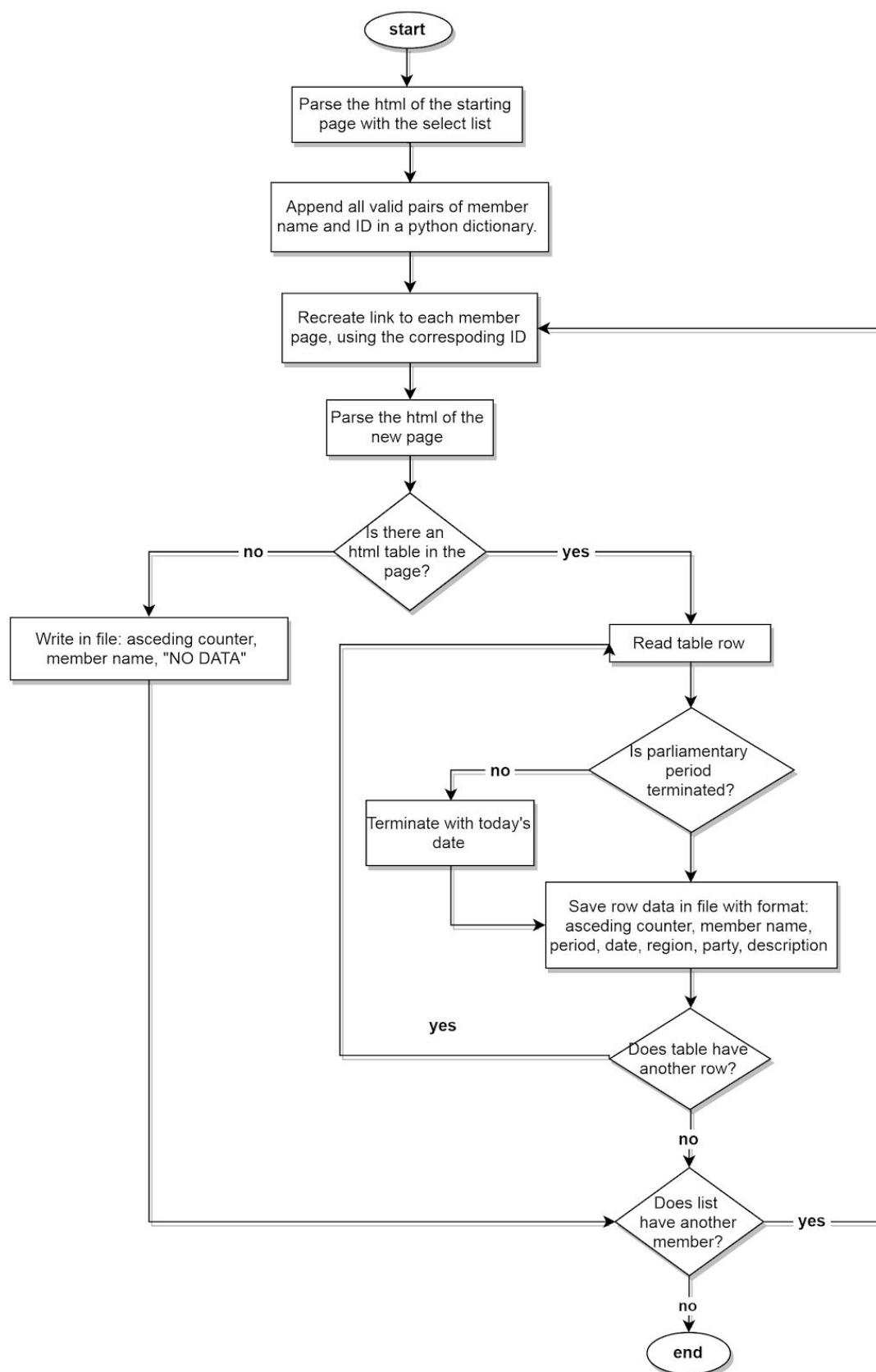


Figure 7: Steps of scraping the data of the Hellenic Parliament members from 1974 until today from the Hellenic Parliament website

4.2 Data Preprocessing

The largest part of the implementation of this project was dedicated to the preprocessing of our data. Below, we provide a detailed description on the challenges we faced and the steps we followed in order to deal with them.

4.2.1 Members Data Cleaning

The members' data that we stored in the file "original_members_data.txt" needed further cleaning and re-formatting in order to be in a more useful format and to include only the data we needed for extracting the desired results. An example of a row in the "original_members_data.txt" file is:

"No:290,Name:Γιαννάκου - Κουτσίκου Παναγιώτη Μαριορή
(Μαριέττα),Period:H'(10/10/1993-24/08/1996),Date:10/10/1993,Administrative-Region:A'A
ΘΗΝΩΝ,Parliamentary-Party:ΝΕΑΔΗΜΟΚΡΑΤΙΑ,Description:Εκλογής"

In this row we can see the counter, the name of the member, the period the person served with exact dates of the period's start and end date in parentheses, a date that refers to the description that follows, the administrative region and party the person belonged to and the description.

The date field falls within the date range of parliamentary period but refers to whatever the description field contains as an event. Usually the description field says "Elected" (in Greek "Εκλογής"). In this case the date field refers to the election day of the parliament member, which coincides with the beginning of the parliamentary period. In the case of someone's resignation during a period, there are 2 entries for that person in the same period, one for their election and one for their resignation, with difference only in the description and date fields. Furthermore, for the person that will replace the resigned member, there will be an entry where the date field will be the day the person started their parliamentary activity and the description will refer to the event of the resignation of the previous member and the



replacement with the new member. In this case, the starting date of the member's activity will not coincide with the beginning of the parliamentary period.

The output file we were aiming to create, would have in each row the following simple format:

“member name, year of activity, political party”

The main problems we had to face were:

- Extracting from a parliamentary period the range of years it included.
- Renaming the political party in a more comprehensible format.
- Restructuring the full name so that we can easily distinguish the first name, last name and possible nickname and discard the father's name, which as we see is available but does not contribute to our future analysis.

For this reason, we created a new script called “members_data_cleaner.py”. This script takes as input the file “original_members_data.txt” and gives as an output the file “members_data.txt”.

In order to achieve this format we had to implemented the following steps:

- We discarded the rows that had the “NO DATA” label.
- We discarded the fields we did not need for our analysis. That is the date, description, and administrative region.
- We converted all member names and political party names in lower case and we removed any accents.
- When a member had two or more first names or last names we joined them with only a dash, without white spaces in between. That way we know that if we split a full name of a member by whitespace, the first part would be the last name(s), the second part would be the father's name, the third part would be the first name(s) and the fourth part in parenthesis would be the nickname, if available.
- The previous step helped us distinguish and remove the father's name from each member.



- We removed extra white spaces and we corrected any mistakes in the names of the members, for example a dash that we found was missing between two first names.
- We extracted from each period the range of years it included and created different entries/rows in the output file for each year of a person. For example “ Period: A'(17/11/1974-22/10/1977)” would include the years from 1974 up to 1977.
- For each of these years we would have to keep only the years from 1989 and after, as the record files we had at our disposal were from 1989 onwards.
- We would have to reform the names of the political parties in a more comprehensible format. For example the party “ΑΝΕΞΑΡΤΗΤΟΙ ΕΛΛΗΝΕΣ ΕΘΝΙΚΗ ΠΑΤΡΙΩΤΙΚΗ ΔΗΜΟΚΡΑΤΙΚΗ ΣΥΜΜΑΧΙΑ” became “ανεξαρτητοι ελληνες εθνικη πατριωτικη δημοκρατικη συμμαχια” and the party “ΟΟ.ΕΟ” became “οικολογοι εναλλακτικοι (ομοσπονδια οικολογικων εναλλακτικων οργανωσεων)”.
- Last but not least, we made sure to remove any duplicate lines.

After this process, from the 1787 unique members that we had at our disposal, we kept information about 1491 unique members. While exploring our dataset of the parliament members, we found that from these 1491 unique members, the 1353 have been in only one political party throughout their career. 111 members have changed between two political parties, 22 members have moved between three political parties, 2 members have changed between four political parties and the remaining 3 members have just served as independent members outside a political party.

4.2.2 Records Cleaning & Data Extraction

To begin with, we converted all downloaded files to simple text file format and translated their file names from Greek to English. In order to do that, we created the python script “file_converter.py” that reads all the files one by one from a directory and moves them to another directory renamed and converted to text.



The main challenge of translating the files from Greek to English was the conversions of the Greek alphabetic numerals to numbers. Greek alphabetic numerals, also known as Ionic, Ionian, Milesian, or Alexandrian numerals, are a system of writing numbers using the letters of the Greek alphabet. Greek numerals are used by the Hellenic parliament proceedings to enumerate the periods, sittings and sessions. Within the Greek numerals, we also found archaic letters of the Greek alphabet that represented numbers. For example the symbol "λ" also known as "sampi" which equals the number 900 and the symbol "Ϟ" also known as "koppa" which equals the number "90" [43]. We also found cases of latin letters written by mistake. For example the letter "a" which was supposed to be written as "α" and the latin letter "p" which was supposed to be written as "ρ".

Apart from the conversion of the Greek numerals to numbers, we translated all the Greek words to English, while trying to keep the special meaning of the parliamentary definitions. For example the string “τμήμα διακοπής εργασιών βουλής θέρους” was all together translated to “-summer-recess-section-”. We also corrected mistakes in the file names. For example we added a space in the string “γ'τμήμα” so that it became “γ' τμήμα” and we could more easily separate the Greek numeral from the word “τμήμα” and achieve the best translation.

After the translation of the file name, we used the tika-app-1.16.jar (<https://tika.apache.org/download.html>) for the conversion of the files to text format. The Apache Tika is a content detection and analysis framework, written in Java, that detects and extracts metadata and text from over a thousand different file types. It has server and command-line editions suitable for use from other programming languages [41]. We spawned the conversion process of the jar file through our Python script with the python library subprocess (<https://docs.python.org/3/library/subprocess.html>). We also kept a log of the renaming process with the name before and the name after in the file “renaming_log.txt”.

The most challenging issue was the conversion of PDF files to text files. That is because the encoding of the PDF records was not always UTF-8. Specifically, 121 PDF files were not designed to correctly contain the way they were encoded. So, even though we tried all the Greek Java supported encodings [48], we extracted garbage every time. There was also a change of undefined encodings over the years. In addition, five pdf files included not text but images of text and required optical character recognition. We decided to exclude these 126



files from our analysis as we considered them of minor importance considering the other 4779 files that we had at our disposal.

Followingly, our main aim was to detect all the speakers and their speeches from the record files and match them with any of the officially reported parliament members from the file “members_data.txt”. For this purpose we created the “member_speech_matcher.py” script. This script takes as input the file “members_data.txt” and gives as an output a file, in which each line has the format: member name, record date, political party, speech.

During this procedure, we tried to minimize the false positives as it was important to have more accurate information, rather than just more information. So, for the matching of the detected speaker names and the actual member names we defined two criteria.

- The first one is a string similarity metric with the use of the Jaro-Winkler [39,40] distance. The Jaro-Winkler distance measures the edit distance between two sequences. It is a variant proposed by W. E. Winkler [40] of the Jaro [39] distance metric and gives more favourable ratings to strings that match from the beginning. The score is normalized so that 0 equates to no similarity and 1 is an exact match. Thus, we determined an acceptance limit of the Jaro-Winkler distance of the two names above 0.95.
- The second criterion we used is that the year of the record in which the speaker was found must match with any of the years that the matched parliament member was active. The latter information is obtained from the file “members_data.txt”, mentioned above.

At this point, it is useful to explain the format of the records. Each record begins with some introductory information such as the date, period, session, sitting, an introductory text, descriptions of procedures. In the figure below you can see a screenshot of the beginning of a record file for the sitting of September 14, 1989.



Π Ρ Α Κ Τ Ι Κ Α Β Ο Υ Λ Η Σ
Ε` ΠΕΡΙΟΔΟΣ (ΠΡΟΕΔΡΕΥΟΜΕΝΗΣ ΔΗΜΟΚΡΑΤΙΑΣ)
Σ Υ Ν Ο Δ Ο Σ Α`
ΣΥΝΕΔΡΙΑΣΗ ΛΖ`

Πέμπτη 14 Σεπτεμβρίου 1989

Αθήνα σήμερα, στις 14 Σεπτεμβρίου 1989, ημέρα Πέμπτη και ώρα 18.46`, συνήλθε στην Αίθουσα συνεδριάσεων του Βουλευτηρίου η Βουλή, σε Ολομέλεια, για να συνεδριάσει υπό την Προεδρία του Προέδρου κ. ΑΘΑΝΑΣΙΟΥ ΤΣΑΛΔΑΡΗ.

ΠΡΟΕΔΡΟΣ (Αθανάσιος Τσαλδάρης): Κύριοι συνάδελφοι, αρχίζει η συνεδρίαση.
(ΕΠΙΚΥΡΩΣΗ ΠΡΑΚΤΙΚΩΝ: Σύμφωνα με την από 13 Σεπτεμβρίου 1989 εξουσιοδότηση του Σώματος, επικυρώθηκαν με ευθύνη του Προεδρείου τα Πρακτικά της ΛΣΤ` συνεδριάσεώς του, της 13ης Σεπτεμβρίου 1989).

Ανακοινώνονται προς το Σώμα από το Βουλευτή κ. Λευτέρη Παπαγεωργόπουλο, τα ακόλουθα:

Α` ΚΑΤΑΘΕΣΗ ΑΝΑΦΟΡΩΝ

1. Ο Βουλευτής Βοιωτίας κ. ΓΕΩΡΓΙΟΣ ΚΑΤΣΙΜΠΑΡΔΗΣ κατέθεσε αναφορά της Πανελλήνιας Ομοσπονδίας Σιδηροδρομικών και της Γενικής Ομοσπονδίας Προσωπικού ΔΕΗ, με την οποία ζητούν την καθιέρωση της απλής αναλογικής σαν πάγιου εκλογικού συστήματος με την κατάργηση του συν 1.

2. Οι Βουλευτές Αιτωλ/νίας κύριοι ΧΡΗΣΤΟΣ ΦΩΤΟΠΟΥΛΟΣ και ΧΡΗΣΤΟΣ ΡΟΚΟΦΥΛΛΟΣ κατέθεσαν αναφορά του Δημάρχου και Μαζικών Θέρμου Νομού Αιτωλ/νίας, με την οποία ζητούν την πλήρη στελέχωση με διδακτικό προσωπικό του Λυκείου Θέρμου.

3. Οι Βουλευτές κύριοι ΔΗΜΗΤΡΙΟΣ ΑΛΑΜΠΑΝΟΣ, ΧΡΗΣΤΟΣ ΦΩΤΟΠΟΥΛΟΣ και ΙΣΑΑΚ ΛΑΥΡΕΝΤΙΔΗΣ κατέθεσαν αναφορά του Αγροτικού Συλλόγου Αταλάντης Νομού Φθιώτιδας, με την οποία ζητεί τη λήψη μέτρων προστασίας του εισοδήματος των καπνοπαραγωγών της περιοχής του.

4. Οι Βουλευτές Αιτωλ/νίας κύριοι ΔΗΜΗΤΡΙΟΣ ΣΤΑΜΑΤΗΣ, ΧΡΗΣΤΟΣ ΦΩΤΟΠΟΥΛΟΣ και ΧΡΗΣΤΟΣ ΡΟΚΟΦΥΛΛΟΣ κατέθεσαν αναφορά του Προέδρου της Κοινότητας Ριγανίου Αιτωλ/νίας, με την οποία ζητεί τη χρηματοδότηση της Κοινότητάς του για την εκτέλεση έργων ύδρευσης.

5. Ο Βουλευτής Μεσσηνίας κ. ΑΡΙΣΤΟΔΗΜΟΣ ΜΠΟΥΛΟΥΚΟΣ κατέθεσε αναφορά του κ. Κων/νου Σακκά, Ταξιάρχου Χωροφυλακής, κατοίκου Αθηνών, με την οποία ζητεί την επανεξέταση των λόγων της αποστρατείας του.

6. Ο Βουλευτής Αιτωλ/νίας κ. ΔΗΜΗΤΡΙΟΣ ΣΤΑΜΑΤΗΣ κατέθεσε αναφορά του

Figure 8: Screenshot of the beginning of a record file of the sitting that took place on September 14, 1989.

The record then continues with the discussion that took place at that sitting. Each speaker's full name is written in full capital letters at the beginning of a new line and is followed by a colon and the corresponding speech, as shown in the screenshot below. Throughout the records, we noticed multiple variations of this pattern, some on purpose and some by mistake, which we will analyze below.



ΠΡΟΕΔΡΕΥΩΝ (Νικόλαος Κατσαρός): Ο Υπουργός Περιβάλλοντος Χωροταξίας και Δημοσίων Έργων, έχει το λόγο.

ΒΟΥΛΕΥΤΕΣ (Από την Πτέρυγα του ΠΑΣΟΚ): Τι σχέση έχει; Έχει σχέση με έργα;

ΠΡΟΕΔΡΕΥΩΝ (Νικόλαος Κατσαρός): Ποιος είπε "τι σχέση έχει" κύριοι συνάδελφοι; Τέλος πάντων, διαβάστε τον Κανονισμό για να μη λέτε αυτά τα απίθανα πράγματα.

ΣΩΤΗΡΗΣ ΚΟΥΒΕΛΑΣ (Υπ. Περιβάλλοντος, Χωροταξίας και Δημοσίων Έργων): Κύριοι συνάδελφοι, οι παρατηρήσεις που έγιναν πάνω στο νομοσχέδιο αφορούν στο χρόνο της εισαγωγής του, στην προετοιμασία του, εάν δηλαδή τώρα είναι ο κατάλληλος χρόνος, εάν προετοιμάστηκε επαρκώς το νομοσχέδιο και η Βουλή προκειμένου να το συζητήσει. Ακούστηκαν απόψεις, για το τι συνεισφέρει κάθε μια από τις Παρατάξεις που συγκροτούν τη Βουλή στα θέματα που το νομοσχέδιο προσδιορίζει, δηλαδή τι συνεισφέρει κάθε Παράταξη στα θέματα της προαγωγής και της απελευθέρωσης των μέσων μαζικής επικοινωνίας και ακόμη τι το ίδιο το νομοσχέδιο προωθεί.

Figure 9: Screenshot of a conversation recorded in the proceedings

In the “member_speech_matcher.py” script we read all records one by one. For each record, we extract the year it was created, from the filename. Then, for each line in the record we try to distinguish whether there is a speaker name at the beginning, followed by a colon and the corresponding speech, with the use of a regular expression. After detection, we try to clean the speaker name with the use of regular expressions, so as to separate any nicknames, roles or other noise. We remove any punctuation, extra tabs and white spaces, dashes, accents, 1 or 2 letter characters, English characters that are used by mistake instead of Greek letters. We, also, turn it to lower case.

At this stage, we have as much of a clean detected speaker name as possible and the year of the record that the speaker was detected in. We are ready to compare it with each entry/line in the official list of parliament members and their years of activity. In order to minimize the running time of the script, we firstly look for entries in the members list that match the record year. In case years match, we proceed with the name comparison using the Jaro-Winkler distance. During the iteration in the comparison loop, we keep in a temporary variable the member name and political party of the maximum Jaro-Winkler distance. After the completion of the comparison with all the official member names, if the maximum Jaro-Winkler distance is over 0.95, we write the information in the “tell_all.csv” file with the following format: matched member name, full record date, political party, speech.



The steps mentioned above are represented in the following figure, where F1 is each record file with the proceedings of that sitting and F2 is the file “members_data.txt” with the officially reported parliament members.



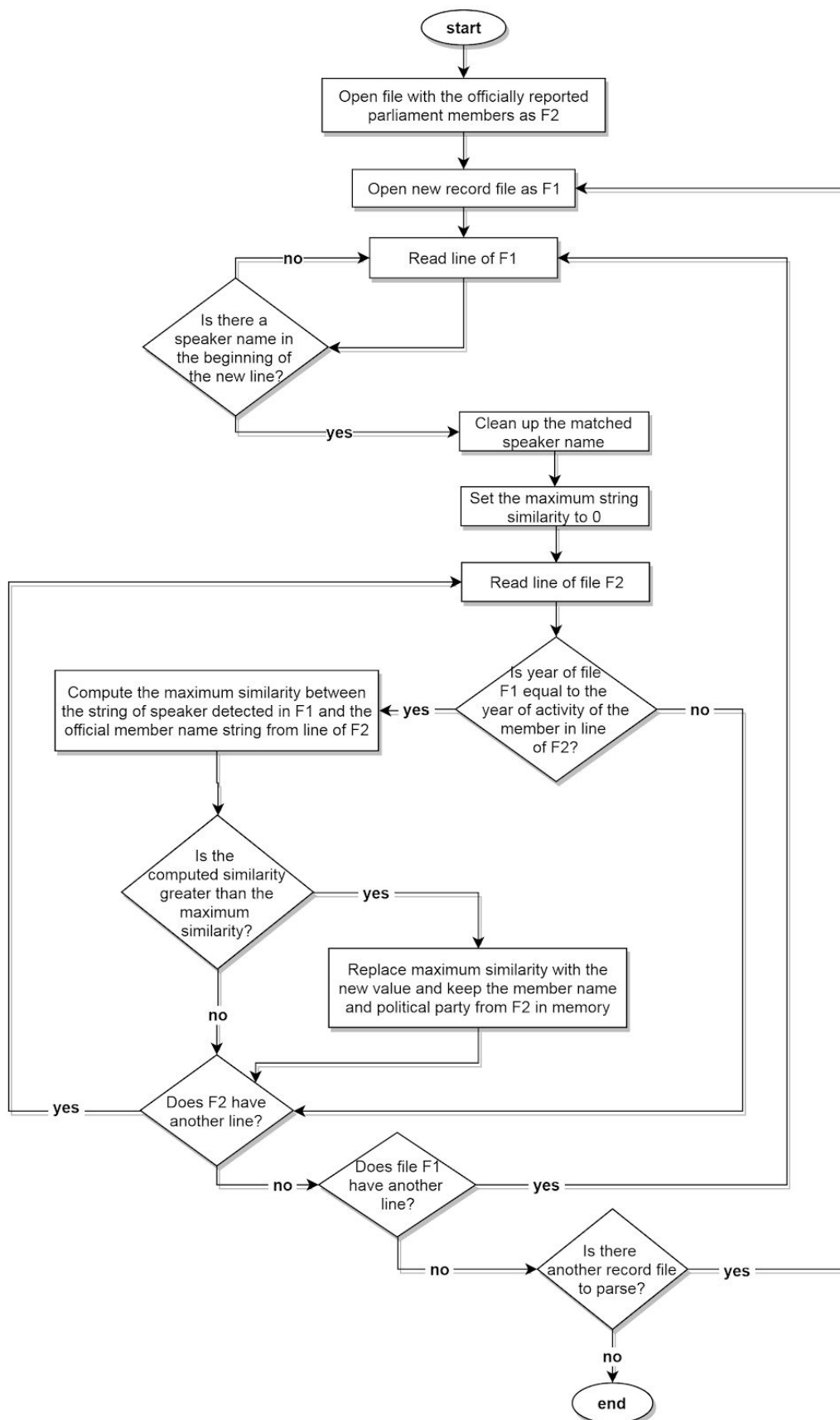


Figure 10: Process of matching the detected speaker of a record file with the official parliament member name

The lack of a specific speaker pattern throughout all records, made it harder for us to detect the actual speakers, their speeches and the identification with the official member names from the “members_data.txt”. The most common difficulties we encountered are listed below:

- **Missing names**

In many cases there is no speaker name but a description in capital letters stating that the speaker is “A parliament member from the XXX political party” or just “A parliament member” or even “Many parliament members”, followed by a colon and the speech. In some other cases, the beginning of the line that declares the speaker consists of the role of the parliament member, for example “SITTING CHAIR OF THE PARLIAMENT”, followed, not always, by the actual full name of the speaker in parenthesis.

To address this, we used regular expressions in order to identify whether a speaker’s name was displayed in parenthesis. In case no name was available but the political party was mentioned before the colon, we used regular expressions to identify it and kept their speech in our “tell_all.csv” file.

In case no political party could also be detected, we kept the speech with a generic reference in our “tell_all.csv” file.

This information, even though not possible to be matched to an actual person, was used in the quality and sentiment analysis for each political party and for all the Parliament in general.

- **Misspelled names**

Sometimes, the speakers’ names had misspellings, missing characters or missing syllables. For this reason, we accepted comparison results with Jaro-Winkler distance less than 1, but always more than 0.95.

- **Name pattern variations**

In many cases, the official names of the parliament members were not used and instead one could find their nickname and surname or first name, nickname in parentheses and surname.



Some speakers in the records appear to have more than one first names or last names. Furthermore, the first name or nickname did not always precede the last name. This comes in contrast with the Jaro-Winkler distance, that takes into consideration the order of the characters in the strings being compared.

To address this situation, we saved the official members' names in the file "members_data.txt" with a structure that facilitated our task, so as to know exactly which is/are the first or last name(s) and nickname(s). Luckily, the official member names list included some nicknames. We also knew from the "members_data.txt" that the members have up to three first names and up to two last names. But, no member has three first names and two last names at the same time. With this information, for each comparison of a detected speaker with an official member's name, we decided to try and construct all the possible ways a member could be referred to by transposing the words of the official member's name in all the possible orders and by interchanging or combining names with nicknames. This gives us the opportunity to raise the accepted Jaro-Winkler distance limit to 0.95 and above, as we exhaust any possible word order in the names.

Below we showcase an explanatory figure of the flow of comparisons. The abbreviation "(SN)", that is included in the figure, represents each "Speaker Name" we detected in the records.



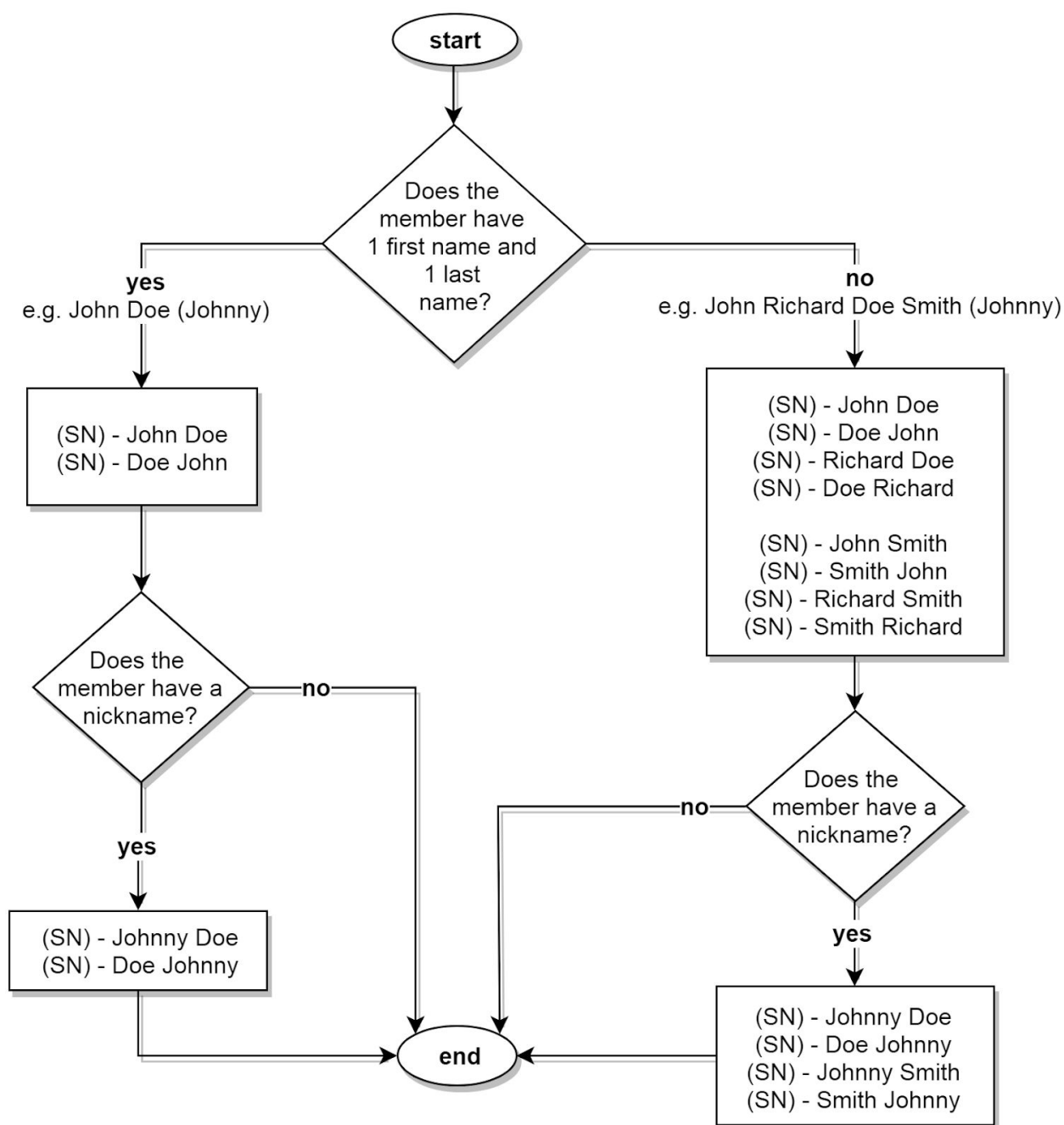


Figure 11: Detailed representation of the string comparison process between the detected speaker in the record file and the official parliament member name

- Poorly formatted text**

On several occasions, the texts with the discussions that took place in the sittings were poorly formatted and did not follow the recommended structure that would help us distinguish speakers and speeches. Sometimes the semicolon that delimits each speaker from the their

speech would be absent, each new speaker-speech would not be separated with the previous speaker-speech by a new line, capital case in speaker names would not be used and much more.

Below you can see an example that comes in contrast with correctly formatted text in Figure 9. In these cases, we performed manual corrections in the “tell_all.csv” file, as we could not implement an automated solution. Thus, we created the file “tell_all_manually_corrected.csv”, that is actually the “tell_all.csv” with manual corrections.



γίνεται Δεκτό,; ΘΕΟΔΩΡΟΣ ΠΑΓΚΑΛΟΣ: Κρατείται. ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Κρατείται και θα συζητηθεί κατά τον Κανονισμό. Μόνη συζήτηση επί της αρχής, των άρθρων και του συνόλου του σχεδίου νόμου: "Κύρωση Συμφωνίας μεταξύ των Κυβερνήσεων της Ελληνικής Δημοκρατίας και της Δημοκρατίας της Αλβανίας για την προώθηση και αμοιβαία προστασία των επενδύσεων". Ερωτάται το Σώμα, γίνεται Δεκτό,; ΘΕΟΔΩΡΟΣ ΠΑΓΚΑΛΟΣ: Κρατείται. ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Κρατείται και θα συζητηθεί κατά τον Κανονισμό. Μόνη συζήτηση επί της αρχής, των άρθρων και του συνόλου του σχεδίου νόμου: "Κύρωση του Πρόσθετου Πρωτοκόλλου που προσαρτάται στη Συμφωνία μεταξύ των Κρατών-Μελών της Ευρωπαϊκής Κοινότητας Άνθρακα και Χάλυβα και της Δημοκρατίας της Ισλανδίας ως και του δεύτερου Πρόσθετου Πρωτοκόλλου μεταξύ των ιδίων Μερών, συνεπεία της προσχώρησης του Βασιλείου της Ισπανίας και της Πορτογαλικής Δημοκρατίας στην Κοινότητα". Ερωτάται το Σώμα, γίνεται Δεκτό,; ΟΛΟΙ ΟΙ ΒΟΥΛΕΥΤΕΣ: Δεκτό, Δεκτό. ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Συνεπώς το νομοσχέδιο "Κύρωση του Πρόσθετου Πρωτοκόλλου που προσαρτάται στη Συμφωνία μεταξύ των Κρατών-Μελών της Ευρωπαϊκής Κοινότητας Άνθρακα και Χάλυβα και της Δημοκρατίας της Ισλανδίας ως και του δεύτερου Πρόσθετου Πρωτοκόλλου μεταξύ των ιδίων Μερών, συνεπεία της προσχώρησης του Βασιλείου της Ισπανίας και της Πορτογαλικής Δημοκρατίας στην Κοινότητα", έγινε Δεκτό, σε μόνη συζήτηση κατ' αρχήν, κατ' άρθρον και στο σύνολο ομοφώνως, έχει δε ως εξής: (Να καταχωρηθεί το κείμενο του νομοσχεδίου) ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Παρακαλώ το Σώμα να εξουσιοδοτήσει το Προεδρείο για την υπ' ευθύνη του επικύρωση των Πρακτικών ως προς την ψήφιση των ως άνω σχεδίων νόμου. ΠΟΛΛΟΙ ΒΟΥΛΕΥΤΕΣ: Μάλιστα, μάλιστα. ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Η Βουλή παρέσχε τη ζητηθείσα εξουσιοδότηση. ΒΙΡΓΙΝΙΑ ΤΣΟΥΔΕΡΟΥ (Υφυπ. Εξωτερικών): Μου επιτρέπετε, κύριε Πρόεδρε; ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Ορίστε, κυρία Υπουργέ. ΒΙΡΓΙΝΙΑ ΤΣΟΥΔΕΡΟΥ (Υφυπ. Εξωτερικών): Κύριε Πρόεδρε, τα 4 σχέδια νόμου που κρατήθηκαν τελικά του Υπουργείου Εξωτερικών, θέλω να σημειώσω ότι πέρασαν ομοφώνως από τη Διαρκή Επιτροπή και θα παρακαλούσα αν μπορούσε να ορισθεί μια μέρα τη μεθεπόμενη εβδομάδα, δηλαδή την τελευταία προ των θερινών διακοπών, για συζήτηση με τη διαδικασία του άρθρου 108. ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Προφανώς, κυρία Υπουργέ, η Διάσκεψη των Προέδρων μεθαύριο, μπορεί να αποφασίσει να περάσουν με τη διαδικασία του άρθρου 108 του καν. όλα αυτά τα νομοσχέδια πριν απ' το καλοκαίρι, γιατί πρέπει να συζητηθούν στην Ολομέλεια. ΒΙΡΓΙΝΙΑ ΤΣΟΥΔΕΡΟΥ (Υφυπ. Εξωτερικών): Μάλιστα, κύριε Πρόεδρε, αλλά θα ήθελα όπως σας είπα και πριν τη μεθεπόμενη εβδομάδα, γιατί την επόμενη θα απουσιάζω. ΠΡΟΕΔΡΕΥΩΝ (Δημήτριος Φράγκος): Μάλιστα. Εισερχόμεθα στη συζήτηση του σχεδίου νόμου: "Κύρωση Συμφωνίας Ελλάδος-Ζαΐρ, για την προώθηση και αμοιβαία προστασία των επενδύσεων". Ο Εισηγητής της Πλειοψηφίας κ. Σαλίκας έχει το λόγο. ΝΙΚΟΛΑΟΣ ΣΑΛΙΚΑΣ: Κύριε Πρόεδρε, κύριοι συνάδελφοι, συζητείται απόψε το σχέδιο νόμου σχετικά με την κύρωση

Figure 12: Screenshot of poorly formatted record for the sitting on June 2nd, 1992.



4.3 Analysis

4.3.1 Introduction

The analysis of our data included speech quality and sentiment evaluation and was implemented in two phases.

Concerning the speech quality analysis, among the various readability formulas available, we selected the SMOG index, as it is considered to be the most widely used. However, as the application of SMOG index on the Greek language lacks statistical validity, we will interpret the results comparatively, without matching them to prerequisite years of schooling [35]. That is, a SMOG index x does not equal x years of schooling; but a SMOG index $x > y$ still indicates that the text with SMOG index y is more difficult to read than the text with SMOG index y .

Furthermore, we implemented sentiment analysis on the records of the Hellenic Parliament, with a fairly simple and efficient method. The sentiment ratings of our corpus are a result of direct calculations derived from the words constructing the corpus, with the use of two sentiment lexicons.

The calculation of the SMOG Index as well as the sentiment scores for each group of texts we selected is implemented with the use of the quadratic mean, also known as root mean square. This type of mean gives a greater weight to larger items in the set and is always equal to or greater than the arithmetic mean.

The two phases of the analysis are the following:

Phase 1: In order to efficiently process the data in the “tell_all_manually_corrected.csv” file, we created a new csv file “tell_all_speaker_per_sitting.csv” with the use of our script “group_speaker_per_sitting.py”. Based on the “tell_all_manually_corrected.csv” file, we grouped together the speeches of each person in each sitting. Each line on the “tell_all_speaker_per_sitting.csv” file has the following format: speaker full name, date, sitting number, party, all speeches of this speaker at that date and sitting concatenated. It was



important to group the speeches in a way that each entry of the csv file would include 30 or more sentences, so that we can apply the SMOG Index calculations.

Then, with the use of “the_analyst.py” script we calculated the speech quality and sentiment of each group of speeches and wrote the results in the new file “tell_all_speaker_per_sitting_analyzed.csv”, that has the following format in each line: speaker full name, date, sitting number, party, SMOG index of grouped speeches, 6-sentiment vector of grouped speeches, number of positive words of grouped speeches, number of negative words of grouped speeches. As we can see, the “grouped speeches” part of each line was replaced with the corresponding metrics.

Phase 2: In this phase, we calculated the quadratic average of SMOG indexes and sentiment vectors as well as the sums of positive and negative words, based on the results we were interested in exporting, with the use of the script “analyst_final_results.py”. The speech quality and sentiment results concerned the evaluation of:

- All parliament speeches throughout all times, as an average reference point.
- All parliament speeches grouped at five-year intervals.
- The speeches of each political party throughout all time.
- The speeches of each political party grouped at five-year intervals.

Below we provide further information on the aforementioned steps and the final results.

4.3.2 SMOG Index Calculator for the Greek Language

While there are some ready-made libraries for the calculation of the SMOG index for the English language, none of these, when applied on the Greek language, would give us correct results. Therefore, we created our own SMOG index calculator for the Greek language, that can be found as the method `smog_index()` in “the_analyst.py” script.

In order to achieve the highest possible accuracy and take into account all the details of the implementation, we firstly created our own SMOG index calculator for the English language and compared the results with the already existing libraries, so as to avoid important



omissions in our approach and achieve the exact same results. Furthermore, we checked the results of the libraries by calculating the SMOG index manually for sample texts of 60 sentences. In these sample texts, we tried to include all the possible cases for sentence and word formations that could confuse our script. For example we included apostrophes, numbers, abbreviations, words connected with dashes, words with different numbers of syllables, very large sentences or very small sentences, sentences that ended with full stop, question mark, exclamation mark, an ellipsis, sentences separated with whitespace, no space or new line.

Among other existing libraries that calculate the SMOG index, the most common are Textstat (<https://pypi.org/project/textstat/>) and Readability (<https://pypi.org/project/readability/>). During the manual check of the libraries, we found that the Readability library would tokenize sentences only when they were separated with '\n'. Even when our sample text would be in this format, the result was not calculated with the exact same widely known SMOG formula, as the results were slightly different. Therefore, we used only the Textstat library for our comparison.

For our English SMOG calculator, we used: the NLTK library (<https://www.nltk.org>) for sentence and word tokenization and the Textstat library for counting syllables.

For the sampling of the sentences of the corpus on which the index calculator would be applied, we splitted the text in three equally sized parts and chose the middle 20 sentences from each part. The size of each part is defined by the number of its sentences. Then, we counted the polysyllables of each text, that is the number of words with more than two syllables. Finally, we applied the formula and verified that our SMOG calculator produced the same results with Textstat library.

During the fine-tuning of our English SMOG calculator and before receiving the aforementioned results, we discovered that the word tokenizer of the NLTK library did not correctly count the syllables of words with their last syllable being solely the letter “-y” such as eas-y, eight-y, speak-eas-y, as-phyx-y. This issue was also present in the SMOG calculator of Textstat library, so we decided to maintained it in our English SMOG calculator in order to achieve the same results and check the validity of our approach.



Furthermore, according to the description of the SMOG Index calculation by G. H. McLaughlin [27], “Any string of letters or numerals beginning and ending with a space or punctuation mark should be counted if you can distinguish at least three syllables when you read it aloud”. However, the Textstat library did not follow this instruction to the fullest as it did not convert digits to text. We did the same in our English SMOG calculator in order to be able to compare our results with those of the Textstat library. However, we implemented this feature in our approach for the Greek language.

For our Greek SMOG calculator, we followed the same approach but replaced the tools for the English language with tools for the Greek language. Specifically, we used the library `greek_accentuation` (<https://github.com/jtauber/greek-accentuation>) for syllabifying the Greek words and the `CLTK` library (<https://github.com/cltk>) for Greek word and sentence tokenization. The Textstat library could not syllabify correctly the Greek words, assuming each Greek word to be one syllable. The CLTK library is only officially supported with Python 3.6 on POSIX-compliant operating systems. We had to make some small amendments, so as to use it on our Windows systems. Furthermore, we used the “wordify_number.py” script for converting digits to text. This script was based on the invoices GitHub project of P. Louridas [38].

Concerning the length of the sample text chosen to calculate the SMOG Index on, we applied the calculation on various sample sizes from at least 30 sentences, as advised in the paper, up to using the whole corpus. We mainly examined and compared the results for 30 sentences, 150, 3000, 12000 and all sentences. From the metrics we extracted in each case, we noticed some fluctuations on the results, mostly on a decimal level, but sometimes reaching 1.5 units. The fluctuations did not have a common pattern. While raising the number of sample sentences, some SMOG indexes would steadily decrease, some would steadily increase, some would first decrease and then increase again and vice versa. The most accurate results should be the results from applying the SMOG Index calculation on the whole corpus that we wanted to analyze each time.

Finally, we introduced our Greek SMOG Index calculator as the method `smog_index()` in “the_analyst.py” script.



4.3.3 Greek Lexicons for Sentiment Analysis

As mentioned above, the sentiment evaluation of our data is based on two sentiment lexicons. We adjusted the lexicons based on our needs and implemented various methods for successfully matching the words of our corpus with the lexicon terms and keeping their sentiments, in an efficient way.

Lexicon 1: Greek Sentiment Lexicon with 6 Sentiment Scores

The first sentiment lexicon we used is a Greek sentiment lexicon that provides ratings for the sentiments of anger, disgust, fear, happiness, sadness and surprise. This lexicon was created with the support of the EC-funded FP7 Project SocialSensor, by Adam Tsakalidis (CERTH-ITI, now University of Warwick) in collaboration with Symeon Papadopoulos (CERTH-ITI) and with the contribution of Ourania Voskaki (Centre for Greek Language) and Kyriaki Ioannidou (Centre for Greek Language) and Christina Boididou (CERTH-ITI) [36].

The lexicon consists of 2,315 Greek terms. In the cases of adjectives, all three genders (male, female, neutral) are implied with the provision of the suffixes (-ος -η -ο). In some cases, the terms refer to components of larger words. Those terms end with hyphen (-).

For the annotation of the data and the provision of each sentiment score, four independent annotators/raters were used. As a result, the lexicon provides four different scores for each sentiment of each term. The possible score values are between 1 and 5. N/A (Not Applicable) value is used in cases that the annotator considers that no value is appropriate for the term or in cases where he/she was not confident on the appropriate value.

The dictionary also contained some linguistic information regarding the entries, as the part of speech, objectivity of each word as evaluated by each annotator and a field with comments that explain the use of the term. The above information is not taken into consideration in this work.

The sentiment lexicon had to be adjusted to the results we were aiming to obtain and to our corpus. Thus, we created the “lexicon_adjuster_6sent.py” script that writes a new sentiment lexicon, namely “out_lexicon_6sent.csv”, based on the Greek Sentiment Lexicon. The string



“6sent” is an abbreviation of the “6 sentiments” that this lexicon provides information for. The script includes the following adjustments:

- For each sentiment we computed the average score of the four annotators, while excluding the “N/A” values from our calculations. For example, in case the sentiment of happiness for a term had values: 1 , 3 , 5 and N/A, we compute the average by adding the first three ratings and dividing by three.
- We populated the lexicon by using the provided suffixes of genders, when available.
- We cleaned entries from multiple whitespaces.
- We removed duplicate entries.
- We removed accents, in order to achieve the best match with the corpus words.
- We removed the terms that refer to components of larger words and end with hyphen (-).
- We removed the terms that had the value “N/A” in all their sentiment ratings.
- We did not apply stemming in our corpus as well as the lexicon, because stemming the lexicon led to multiple identical entries with different scores. Specifically, applying stemming on the 2315 lexicon entries, resulted in 457 common entries. From now on we will refer to this lexicon as Lexicon 1.

Lexicon 2: Greek Sentiment Lexicon with Positive & Negative Sentiment scores

The second sentiment lexicon we used is a lexicon we created by combining words and their sentiment from two files. The two files were two separate lists of positive and negative Greek words [37], created under the “MultilingualSentiment” project of the Data Science Lab of the Stony Brook University of New York. This project includes international open source sentiment lexicons, licensed under the GNU General Public License.

With the use of our “lexicon_adjuster_pos_neg.py” script, we created the file “out_lexicon_pos_neg.csv”. Each row in this file has one of the Greek terms found in the files and the symbol “+” or “-” depending on the file it came from. It includes 2701 Greek terms, of which the 1066 are positive terms and the 1635 are negative terms. Accents have been removed from the terms of the final lexicon, in order to achieve higher matching scores during identification. From now on, we will refer to this lexicon as Lexicon 2.



4.3.4 Word Identification

The identification of a corpus word with a lexicon term was a demanding and time consuming process, especially due to the size of the corpus and its specialized vocabulary. In order to achieve the most accurate matches in an efficient way, we utilized the following tools:

- Our most important criterion for identifying the lexicon entries in the corpus was the Jaro-Winkler distance. We computed the Jaro-Winkler distance between each corpus word and lexicon term and accepted the matches with a distance equal to or above 0.97. We did not limit the distance only to 1.0 (exact match), as we wanted to match the words of our corpus that are in plural with the lexicon terms that are not available in plural.
- While exploring the parliament corpus, we noted that its specialized vocabulary posed a threat to the validity of our identification process with the use of Jaro-Winkler distance. For example, the word 'κύριε' (sir) which was the fourth most common word with 311,637 appearances would match the Lexicon 1 entry 'κυριευω' (capture). This and such other cases could easily affect the sentiment vectors.

To address this issue, we aimed to manually check the matches of the 400 most common corpus words with the Lexicon 1 and Lexicon 2 entries. This procedure resulted in the creation of two files, one for each lexicon, with the manually accepted and rejected matches. So, during the identification process of a corpus word with each lexicon term we first check if this corpus word belongs to the top 400 most common words. If it does, we use the manually provided information and we do not look for a match in the lexicon files.

For implementing this approach, we created the “freq_counter.py” script that gives as an output the “word_frequencies.csv”. This file includes all the unique words of the parliament corpus and their frequencies, sorted with descending order. For this task, we removed any accented letters and some of the punctuation of the corpus. We did



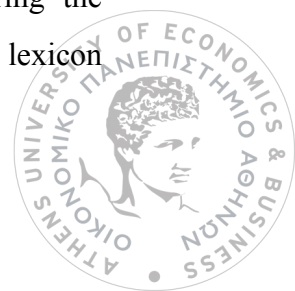
not remove acute accents that, when next to a single letter, suggest Greek numerals such as “α’ sitting”.

Then, with the use of our “top400words.py” script, we read the top 400 most common words of the “word_frequencies.csv” file and we try to match them with the lexicon entries of each lexicon. For words with less than 5 characters, we set the Jaro-Winkler distance accept limit to 0.97 while for longer words we applied the limit of 0.91. This script gives two output files, the “top400_bestmatch_6sent.csv” file corresponding to Lexicon 1 and the “top400_bestmatch_pos_neg.csv” file corresponding to Lexicon 2. Each row of these files had the following information: the corpus word under investigation, the lexicon term that best matched the corpus word, the sentiment scores of the lexicon term and an acceptance index (a yes/no field to be manually edited).

Due to the fact that many matches were not correct, we manually edited the acceptance field and rejected the cases that matched unsuccessfully with a lexicon term. The final manually edited files are namely:

- “top400_6sent_manually_accepted.csv” and
 - “top400_pos_neg_manually_accepted.csv”
- The comparison of each corpus word with each lexicon term was highly time and resource consuming, as we had to iterate through the lexicons and calculate the Jaro-Winkler distance as many times as the corpus words that we wanted to identify.

In order to minimize the calculations and iterations, we used the Pygtrie, a Python library that implements a trie data structure (<https://github.com/google/pygtrie>). This structure, also known as radix or prefix tree, is a tree associating keys to values where all the descendants of a node have a common prefix (associated with that node) [41]. So, we create the trie for each lexicon and we search more efficiently for a match in the lexicon. In addition to that, we have enriched the searching process with one more criterion that has to do with the length of the words. Specifically, during the identification process of a word, we check if the first half word is listed in the lexicon



trie. If so, we iterate through the trie sub-items of the first half word. If the length of the trie sub-item is equal to or smaller than two times the length of the first-half word plus four, only then do we calculate the Jaro-Winkler distance. That way we minimize the searching time and we avoid unnecessary calculations.

4.3.5 Mathematical formula of Sentiment Evaluation

In general, we represent the sentiment of each input text by one sentiment vector of 6 sentiment components by Lexicon 1 and two indexes (positive and negative) by Lexicon 2. The evaluation is implemented in two phases, as mentioned in the section “4.3.1 Introduction”. Below, we explain the mathematics we used for extracting the sentiment for each given text.

In Phase 1, in the script “the_analyst.py”, we compute the sentiment metrics of each speech of the file “tell_all_speaker_per_sitting.csv” and we write the results in the file “tell_all_speaker_per_sitting_analyzed.csv”. To do so, we perform the following steps:

1. For each word of a speech that matched a Lexicon 1 entry, we form one vector with 6 components, one for each examined sentiment from the Lexicon 1. We then have N vectors W_j .

$$\overline{W}_j = [w_{1_j}, w_{2_j}, w_{3_j}, w_{4_j}, w_{5_j}, w_{6_j}]$$

where $j = 1 \dots N$ and N is the number of Lexicon 1 entries identified in the text.

2. We then form a 6 component vector T for the speech:

$$\overline{T} = [t_1, t_2, t_3, t_4, t_5, t_6]$$

of which, each component “t” is a result of the following quadratic mean formula:



$$t_i = \sqrt{\frac{\sum_{j=1}^N w_{i_j}^2}{N}} \quad i=1..6 \quad (1)$$

where i is the number of components of vector T . Formula (1) is the quadratic mean of lexicon entries that were identified in the text.

3. For Lexicon 2, we simply compute the sum of positive and the sum of negative ratings of all the words of the speech that matched a Lexicon 2 entry.

In Phase 2, in the script “analyst_final_results.py”, for Lexicon 1, we re-compute the quadratic mean of each sentiment for the selected T vectors, according to the mathematical formula (1). For Lexicon 2, we compute the sum of the selected sums of positive and negative words. By “selected” we mean those metrics that refer to the speeches that meet the requirements of our query e.g. the sentiment evaluation of the political party “ANEL - Panos Kammenos” for the years 2009-2013.

For Lexicon 2, as the simple sum of the positive and negative words can be influenced by the total amount of the words of a text, we also count the total amount of words for each of the queries we investigate, with our script “count_words.py”. We write the results in the file “count_words_results.txt” and we use these results in the file “pos_neg_result_graphs.py”, which we mention in the section “5. Results” and is responsible for the creation of the graphical animation of the results. In this script, we calculate and represent in graphs the percentage of positive and negative words for each given text based on its total amount of words.



4.3.6 Mathematical formula of Readability Evaluation

The calculation of the SMOG Index is implemented in two phases, as mentioned in the section “4.3.1 Introduction”. Below, we explain the mathematics we used for extracting the SMOG Index of a given text.

In Phase 1, in the script “the_analyst.py”, we compute the SMOG Index of each speech of the file “tell_all_speaker_per_sitting.csv” and we write the results in the file “tell_all_speaker_per_sitting_analyzed.csv”. To do so, we use the SMOG formula already mentioned in section “3.2.3 SMOG Index”:

$$\text{grade} = 1.0430 * \sqrt{\text{number of polysyllables} * \frac{30}{\text{number of sentences}}} + 3.1291$$

In Phase 2, in the file “analyst_final_results.py”, we compute the quadratic mean of the selected SMOG Indexes.

4.3.7 Summary of Analysis Implementation

Phase 1: Summing up all the above preparation steps and the mentioned goals, we describe the steps we follow for the implementation of the speech quality and sentiment analysis of the file “tell_all_speaker_per_sitting.csv” with the use of “the_analyst.py” script. Figure 13 below consists the graphical representation of the process.

1. We load the “top400_6sent_manually_accepted.csv” file in the form of a python dictionary. This file refers to the manually accepted or rejected matches of the top 400 words with the terms of Lexicon 1.
2. We load the Lexicon 1 in a trie data structure.
3. We load the “top400_pos_neg_manually_accepted.csv” file in the form of a python dictionary. This file refers to the manually accepted or rejected matches of the top 400 words with the terms of Lexicon 2.
4. We load the Lexicon 2 in a trie data structure.
5. For each line in the “tell_all_speaker_per_sitting.csv”, we extract the speech part.



6. We calculate the smog index of the speech.
7. We calculate the sentiment vector of the speech based on Lexicon 1. For each word of the speech we do the following:
 - a. We search if the word is listed in the “top400_6sent_manually_accepted.csv” file of Lexicon 1. If it does, we check if we accept the match or not. If we accept it, we append the sentiments of the match in a list and proceed to the next word. If we reject the match, we proceed straight away to the next word.
 - b. If the word is not in the “top400_6sent_manually_accepted.csv” file, we extract the first half of the word and check if this first half is listed in the Lexicon 1 trie as a subtrie. If not, we proceed to the next word. If the first half word is listed in the Lexicon 1 trie, we iterate through the trie sub-items of the first half word. If the length of the trie sub-item is equal to or smaller than two times the length of the first-half word plus four, we calculate the Jaro-Winkler distance. If the Jaro-Winkler distance is equal to or larger than 0.97 we append the sentiments of the Lexicon term match in a list.
 - c. After we appended in the list the sentiments of all the successful matches, we compute the quadratic mean of each sentiment for the speech.
8. We calculate the sentiment vector of the speech based on Lexicon 2. For each word of the speech we do the following:
 - a. We search if the word is listed in the “top400_pos_neg_manually_accepted.csv” file of Lexicon 2. If it does, we check if we accept the match or not. If we accept it, we append the sentiment of the match in a list and proceed to the next word. If we reject the match, we proceed straight away to the next word.
 - b. If the word is not in the “top400_pos_neg_manually_accepted.csv” file, we extract the first half of the word and check if this first half is listed in the Lexicon 2 trie as a subtrie. If not, we proceed to the next word. If the first half word is listed in the Lexicon 2 trie, we iterate through the trie sub-items of the first half word. If the length of the trie sub-item is equal to or smaller than two times the length of the first-half word plus four, we calculate the Jaro-Winkler distance. If the Jaro-Winkler distance is equal to or larger than 0.97 we append the sentiment of the Lexicon term match in a list.



- c. After we appended in the list the sentiments of all the successful matches, we compute the sum of the positive sentiments and the sum of the negative sentiments.
9. We replace the initial speech with the smog index, the quadratic mean of the sentiment vectors and the sums of the positive and negative words. If all metrics are equal to 0, we discard the whole entry/line of the “tell_all_speaker_per_sitting.csv” and we proceed to the next line.
10. We write the results in the new file “tell_all_speaker_per_sitting_analyzed.py”.

In the figure 13 below you can see the basic steps for the calculation of the sentiment vector for each speech, regardless of the Lexicon used. It is an abstract graphical representation of the step 7 as well as the step 8.



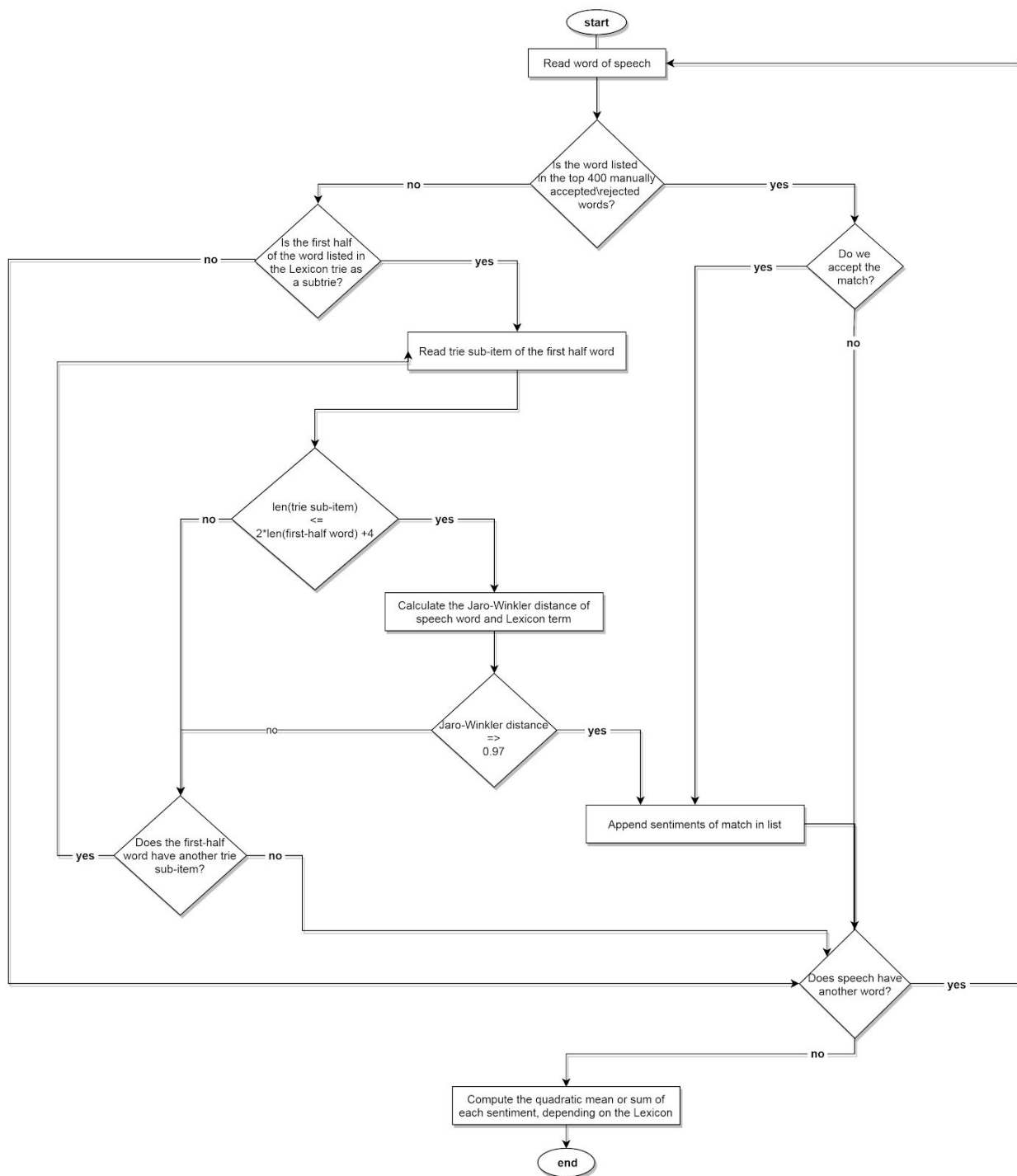


Figure 13: Representation of the basic steps for the calculation of the sentiment vector for each speech

Phase 2: As mentioned above, in this phase, we calculate the quadratic average of SMOG indexes and sentiment vectors as well as the sums of positive and negative words, based on the results we were interested in exporting, with the use of the script “analyst_final_results.py”.

We use python dataframes for efficiently extracting the speeches of specific dates and/or parties we are interested in and we write the results in the file “results.txt”.

5. Results

The results were produced by the scripts “smog_result_graphs.py”, “six_sentiments_result_graphs.py”, “pos_neg_result_graphs.py”. In these scripts we used the NumPy (<http://www.numpy.org>) and Matplotlib (<https://matplotlib.org>) python libraries.

5.1 Speech quality results

The figure below shows the average SMOG Index from 1989 until 2017. We implemented the SMOG Index calculations on speeches grouped as follows: 1989-1993, 1994-1998, 1999-2003, 2004-2008, 2009-2013, 2014-2017. For the representation of the results in the form of a graph, we connected the middle year of each year-group with its average SMOG Index. These pairs are represented below as the dots on the blue line. The grey line represents the all time average SMOG Index of the parliament records. As we can see, there is a great decrease in the average SMOG index, and thus in the speech quality, of the parliament from the early 00s until 2017.



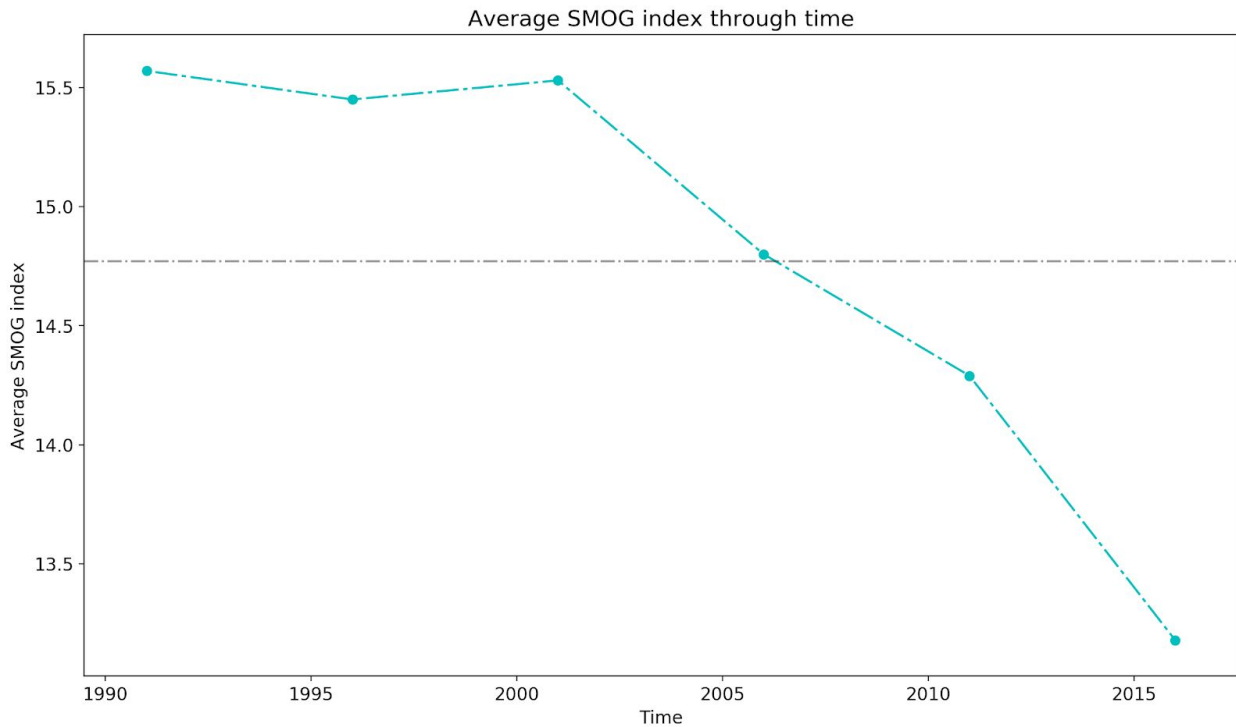


Figure 14: Average SMOG Index score of all parliament records from 1989 until 2017

The following graph represents the measurements for each five-year interval and each political party. The dots in the graph connect the middle year of each time interval with the corresponding SMOG Index of the five-year calculation. In the graph we can only see the political parties whose presence spans between at least two intervals, so that two dots are connected with a line. The dashed grey line represents the average SMOG Index for each interval.

As we can see, there is a generalized decrease in the SMOG Index since the early 2000s. The downward trend is stronger for the political parties DIMAR and Golden Dawn, which show the lowest SMOG Index in the recent years during which they were elected. While most political parties show a decreasing SMOG Index, we can see that LAOS and SYRIZA show an increase in the recent years. However, they are still lower than other political parties such as PASOK and ND, and in general lower than the average SMOG Index, represented by the dashed grey line.

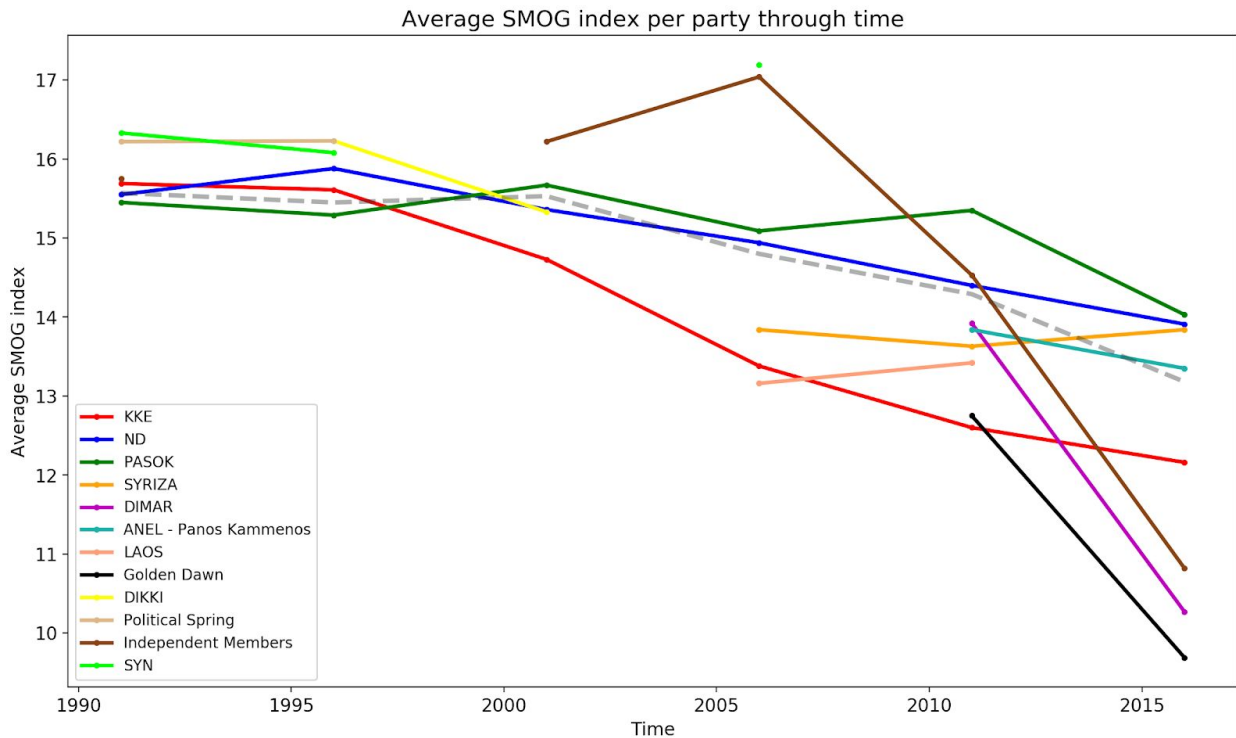


Figure 15: Average SMOG Index score of each political party from 1989 until 2017

In the next figure we can see the average SMOG Index for each political party throughout all years from 1989 until 2017. The faint grey line shows the all time average SMOG Index of all the parliament records and as we can see the majority of the political parties is below the average SMOG Index. The lowest speech quality is held by the political parties “Union of Centrists”, “Golden Dawn” and “ANEL - National Patriotic Democratic Alliance” (not to be confused with ANEL - Panos Kammenos political party). The highest speech quality is held by the political parties “Democratic Social Movement” (DIKKI), “Alternative Ecologists” and the “Coalition of the Left, of Movements and Ecology” (SYN).

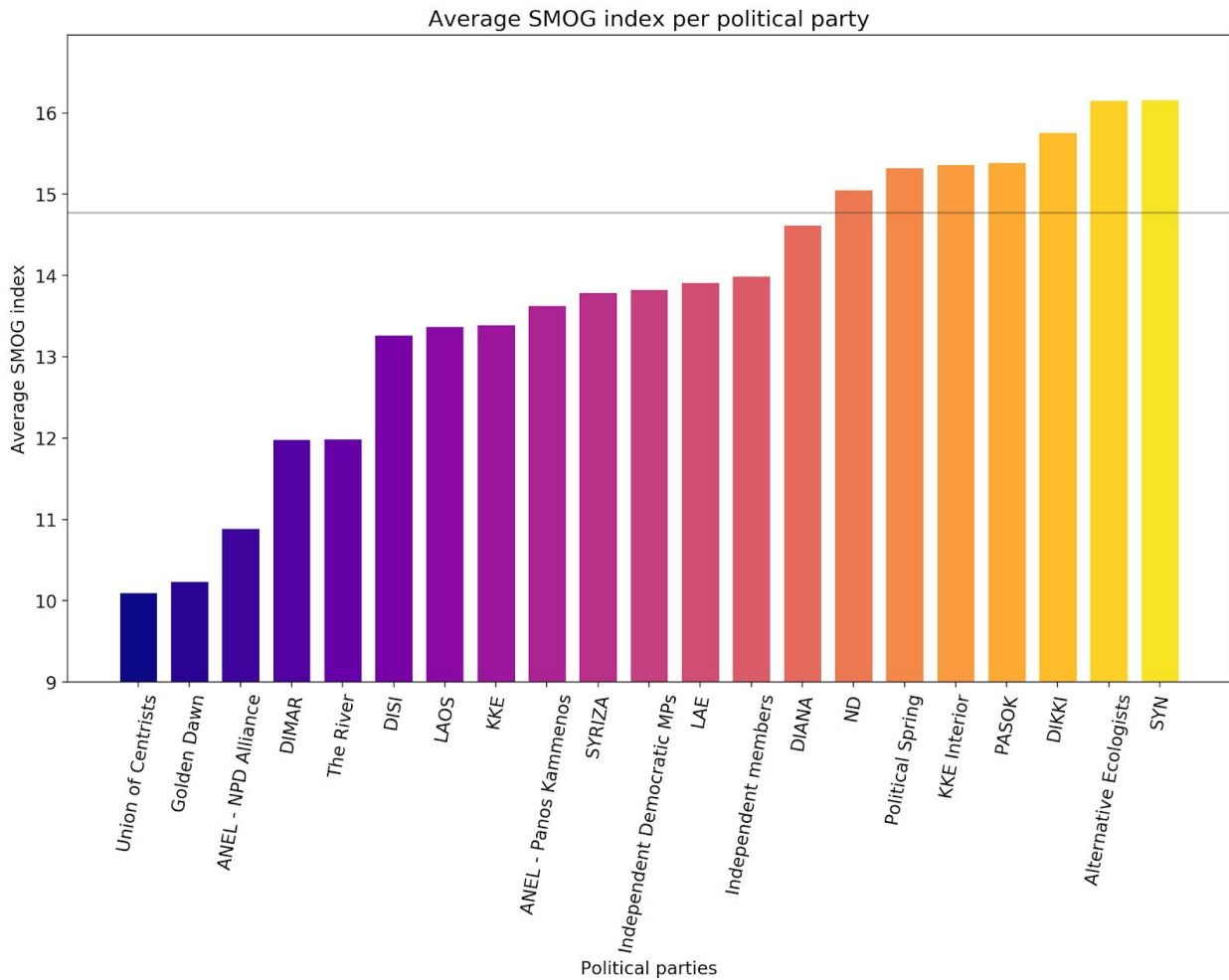


Figure 16: Average SMOG Index score of each political party during all years from 1979 until 2017

5.2 Sentiment analysis results based on Lexicon 1

The following figure represents the average sentiment score for six different sentiments, namely anger, disgust, fear, happiness, sadness and surprise, for the years 1989-2017. The sentiment scores can take values from 0 to 5. The sentiments with the higher score are surprise followed closely by anger and disgust. This means that the speeches and discussions that take place in parliament sittings are mainly characterized by these sentiments. The less common sentiment encountered is sadness. Happiness and fear fall in between the most common sentiments and sadness. The score of the sentiments is quite steady throughout the years.

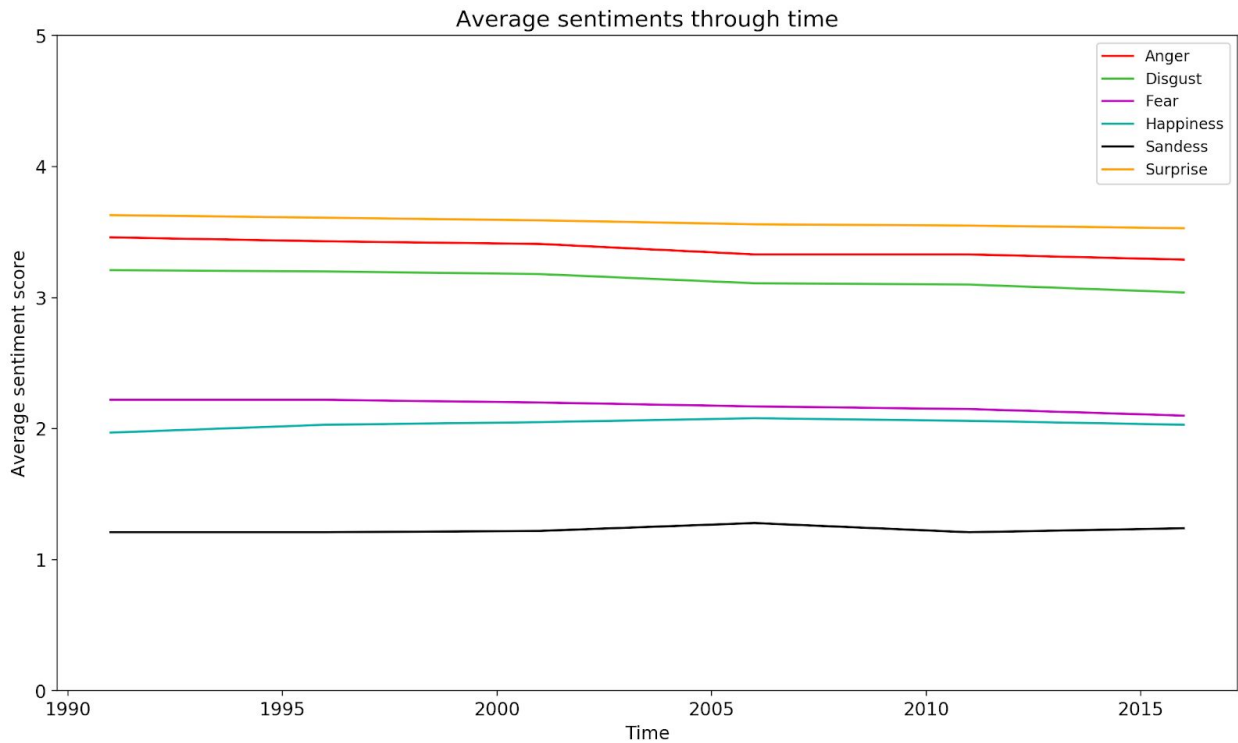


Figure 17: Average sentiment scores for all parliament records from 1989 to 2017

In the next figures we can see the average sentiment scores of each political party throughout the years. Most of the political parties have very similar scores for each sentiment. The difference between the lowest and the highest score for each sentiment does not exceed 0.20 units. The higher scores of anger, disgust and surprise are represented by KKE interior. The higher score of fear is held by the “Democratic Social Movement” (DIKKI), the highest score of happiness by the “Popular Orthodox Rally” (LAOS) and the highest score of sadness by the “Union of Centrists”.



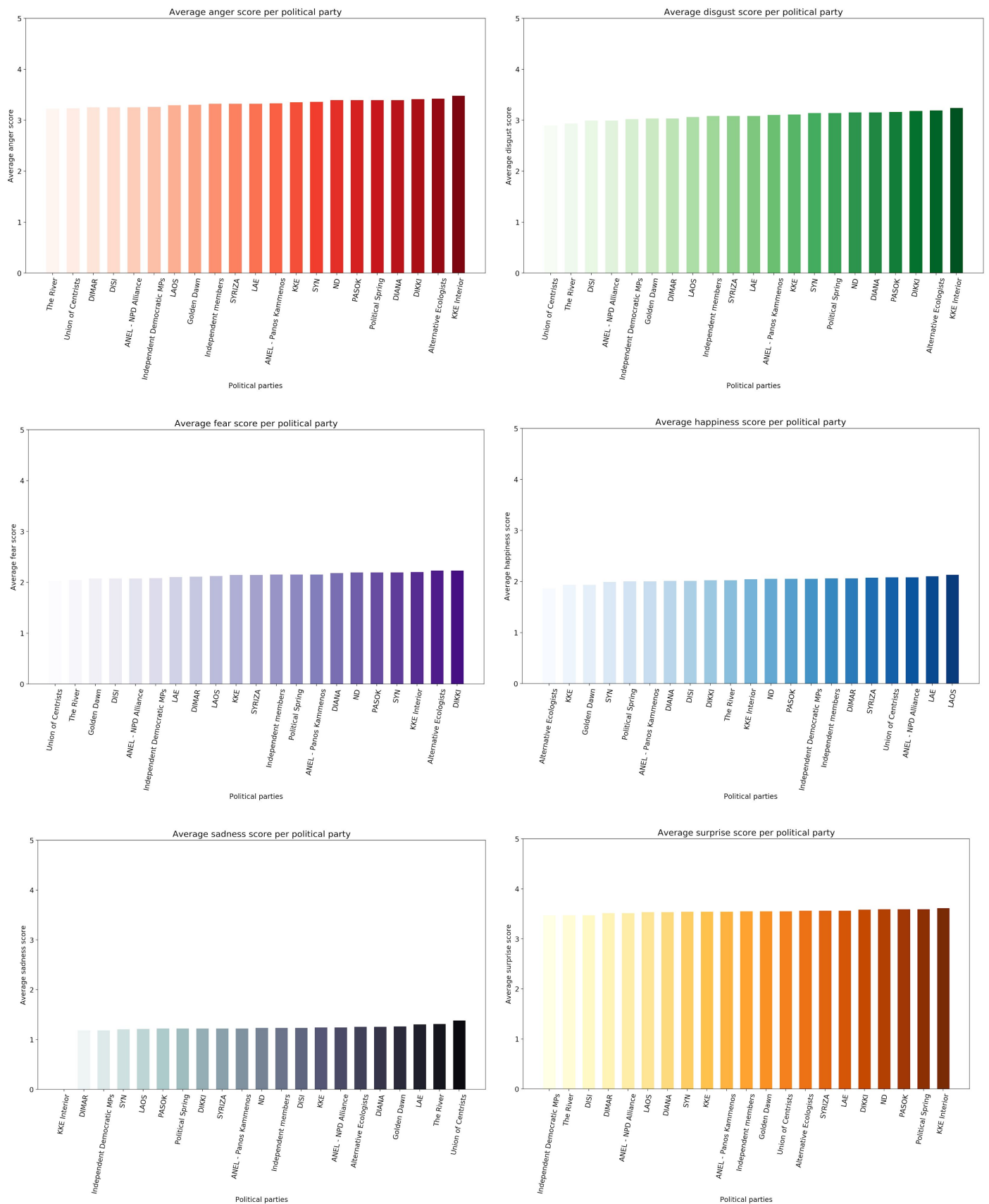
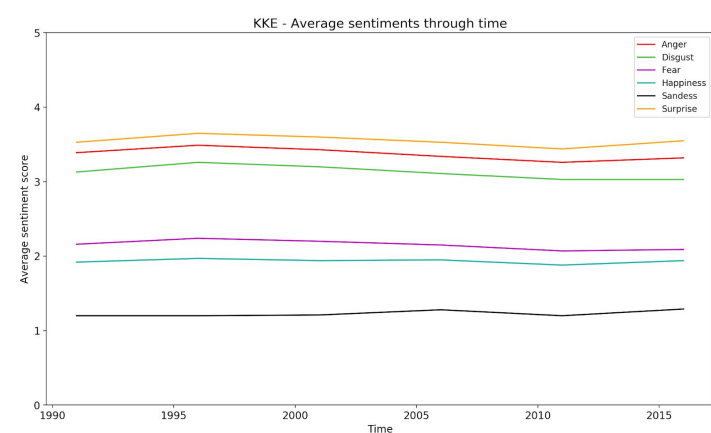
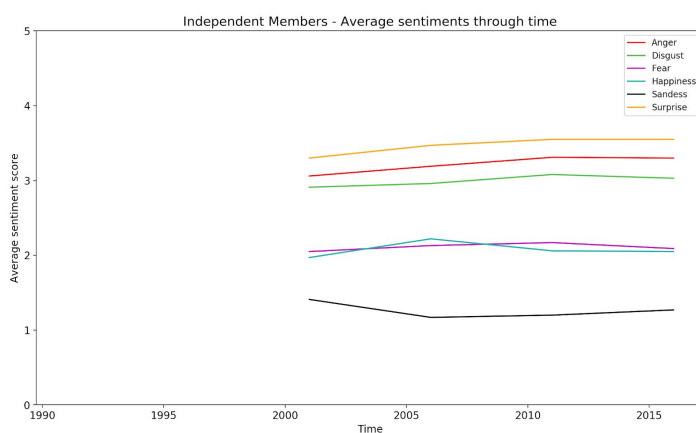
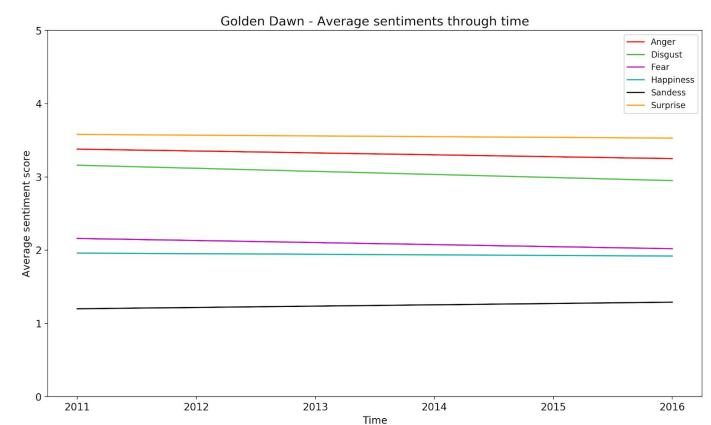
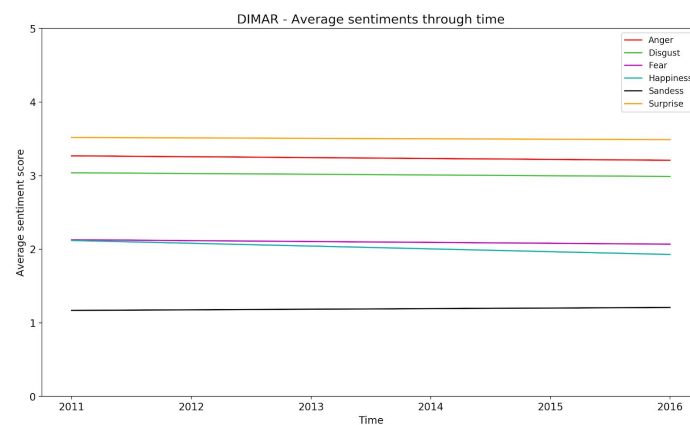
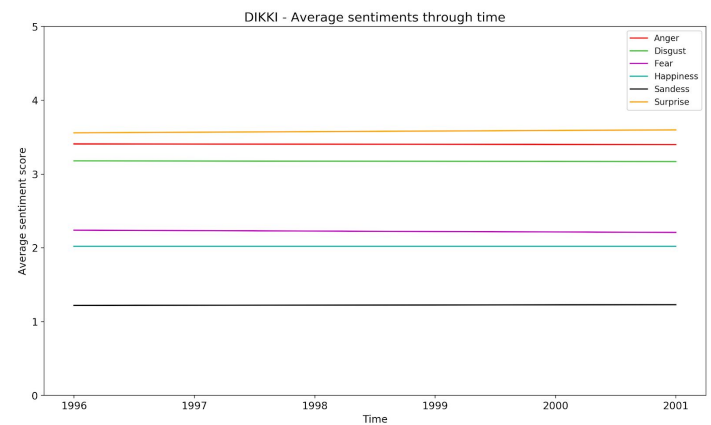
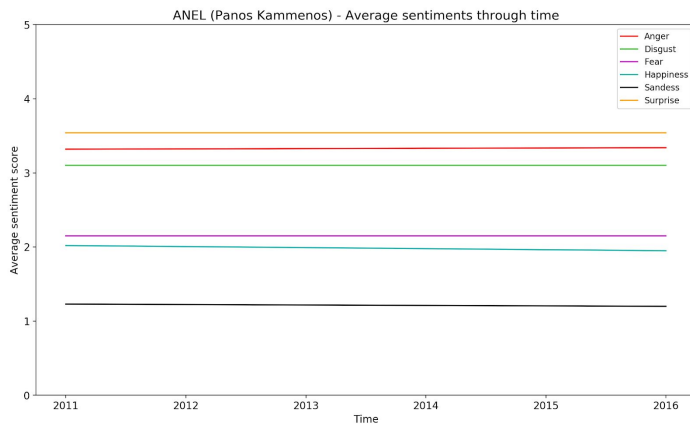


Figure 18: Average sentiment score for each of the 6 sentiments and for each political party during all years from 1989 until 2017.

The following figures showcase the average sentiment scores of each political party from 1989 up to 2017. The scores are quite steady throughout the years and follow the average sentiment scores of all parliament records.



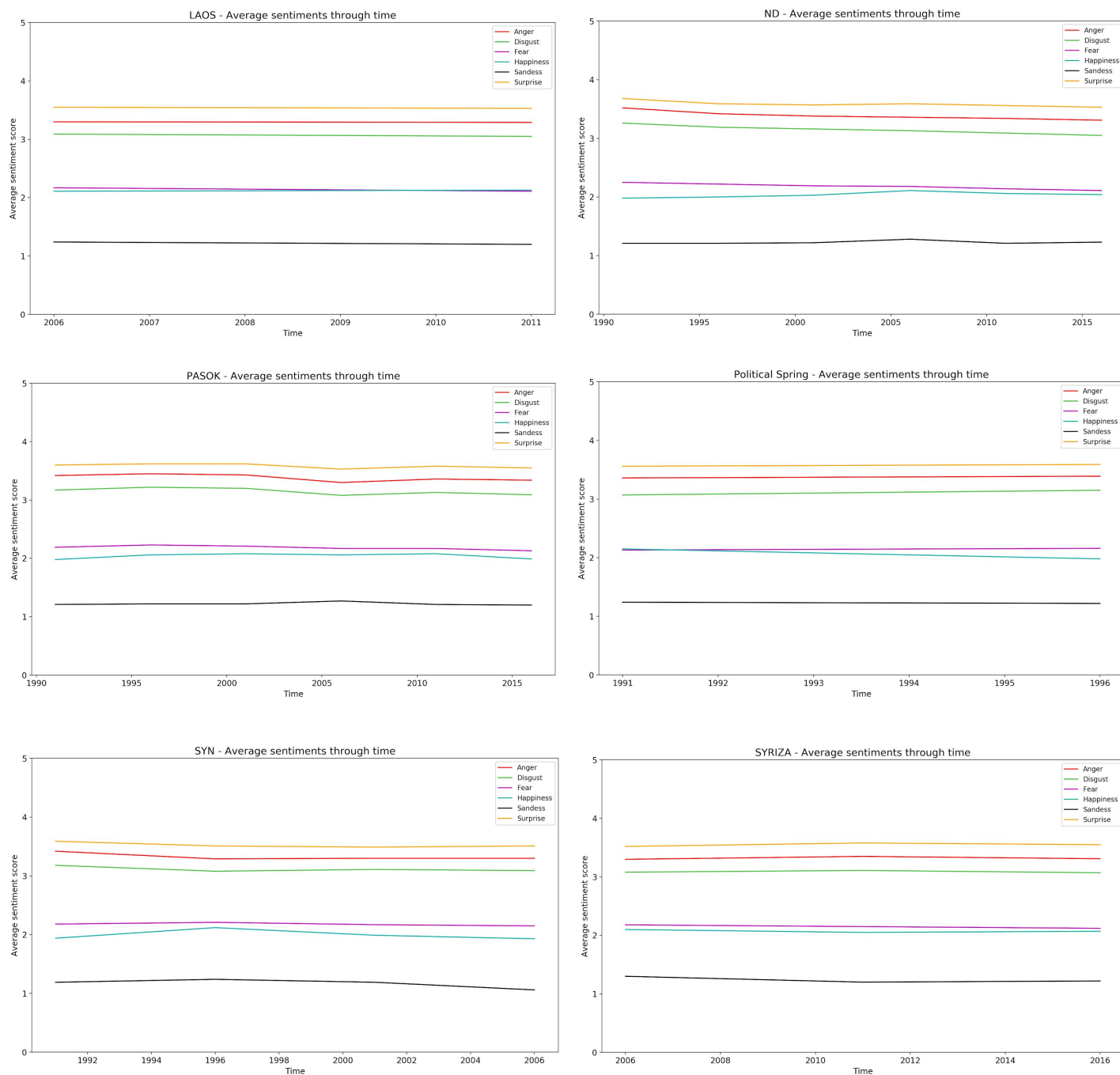


Figure 19: Average sentiment scores for each political party from 1989 to 2017.

5.3 Sentiment analysis results based on Lexicon 2

The following figure represents the percentage of positive and negative words found in the records of the Hellenic Parliament from 1989 until 2017, based on Lexicon 2. The filled green and red lines show the percentage of positive and negative words accordingly, for the measurements have been taken for the following intervals: 1989-2003, 2004-2008, 2009-2013, 2014-2017. In the figure, each interval is represented by a dot that connects the average percentage of positive or negative words with the middle year of that interval. The dashed faint green and red lines show the average percentage of all positive and negative words accordingly, throughout all times. As we can see, the positive words have a stronger presence than negative words during all the years that we examined. In recent years, the graph shows an increase in the percent number of positive words while the negative words remain around their average all-time percentage. This means that the speeches and discussions that take place in parliament sittings are more positively than negatively charged, especially in the last decade.

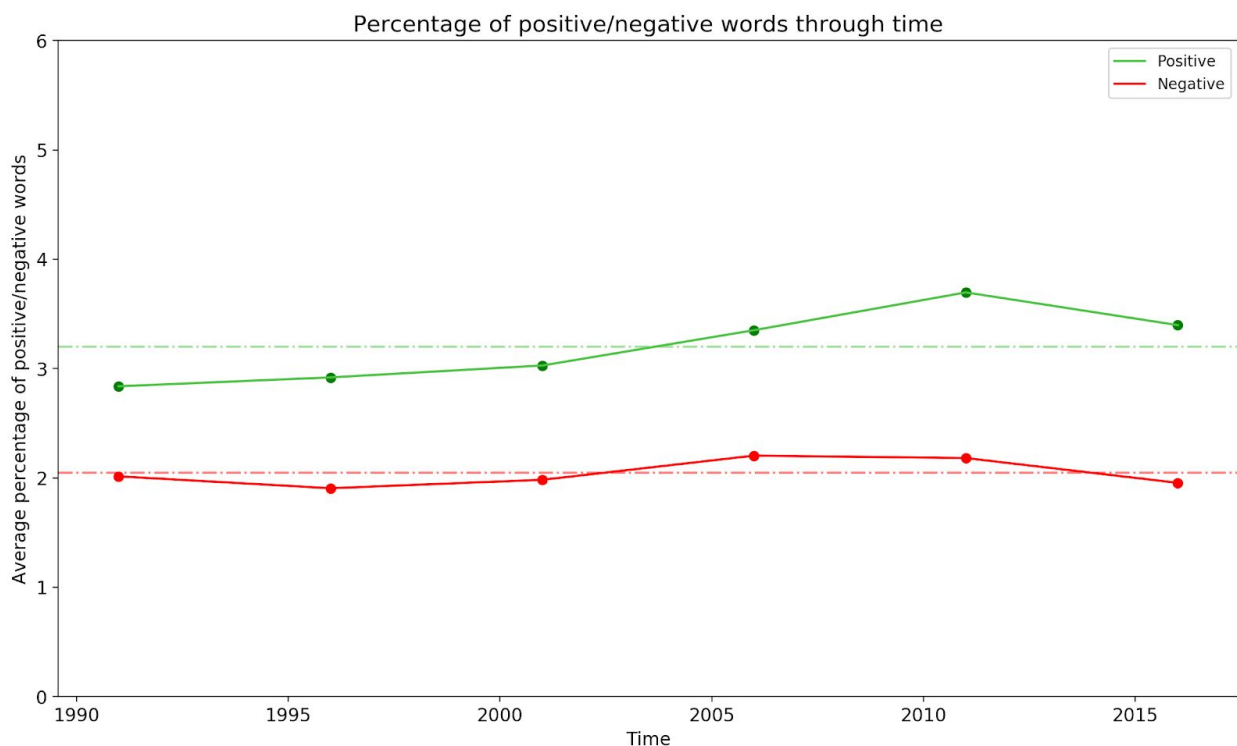
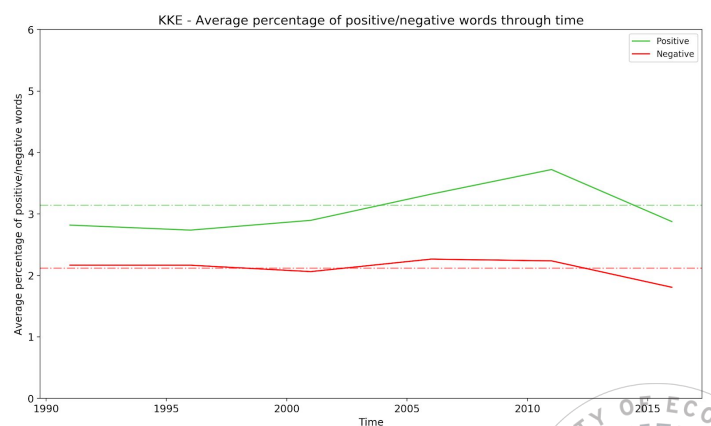
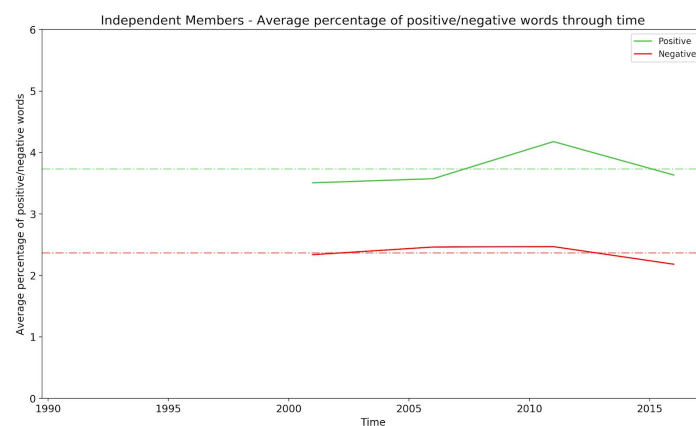
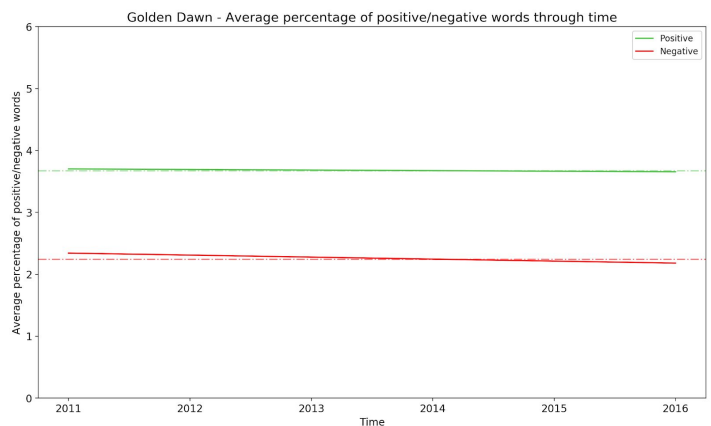
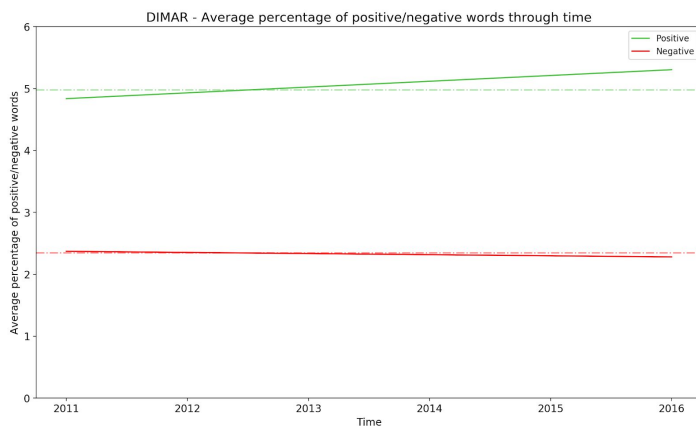
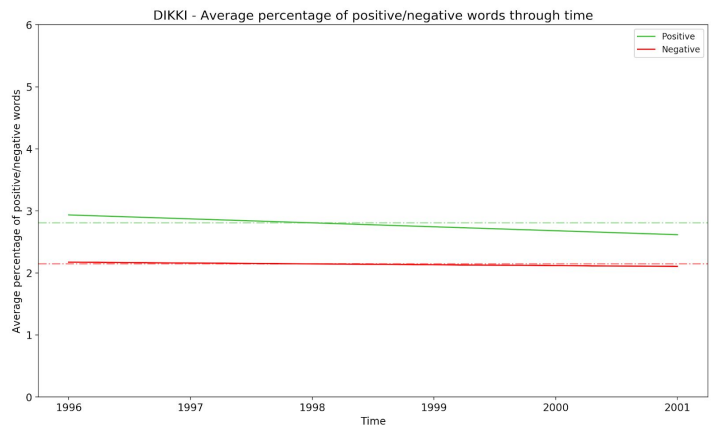
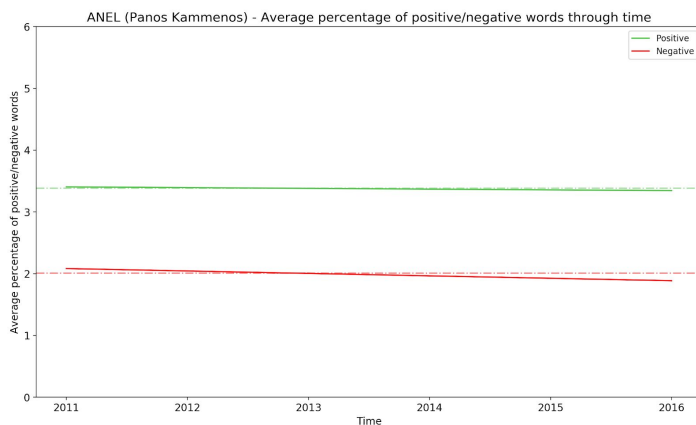


Figure 20: Average percentage of positive and negative words in all the parliament records from 1989 until 2017



Below we display a set of graphs that represent the percentage of positive and negative words we measured for the aforementioned year intervals and for each political party. Once again, we present graphically the measurements of political parties with a presence spanning over at least two different year intervals. In all the graphs, we can see the positive words prevailing over the negative words for all political parties during all years. In most cases we observe that there is an increase in the percentage of positive words in the last decade, followed in some cases by a smaller decrease, leaving the average percentage of positive words higher than in the decade before.



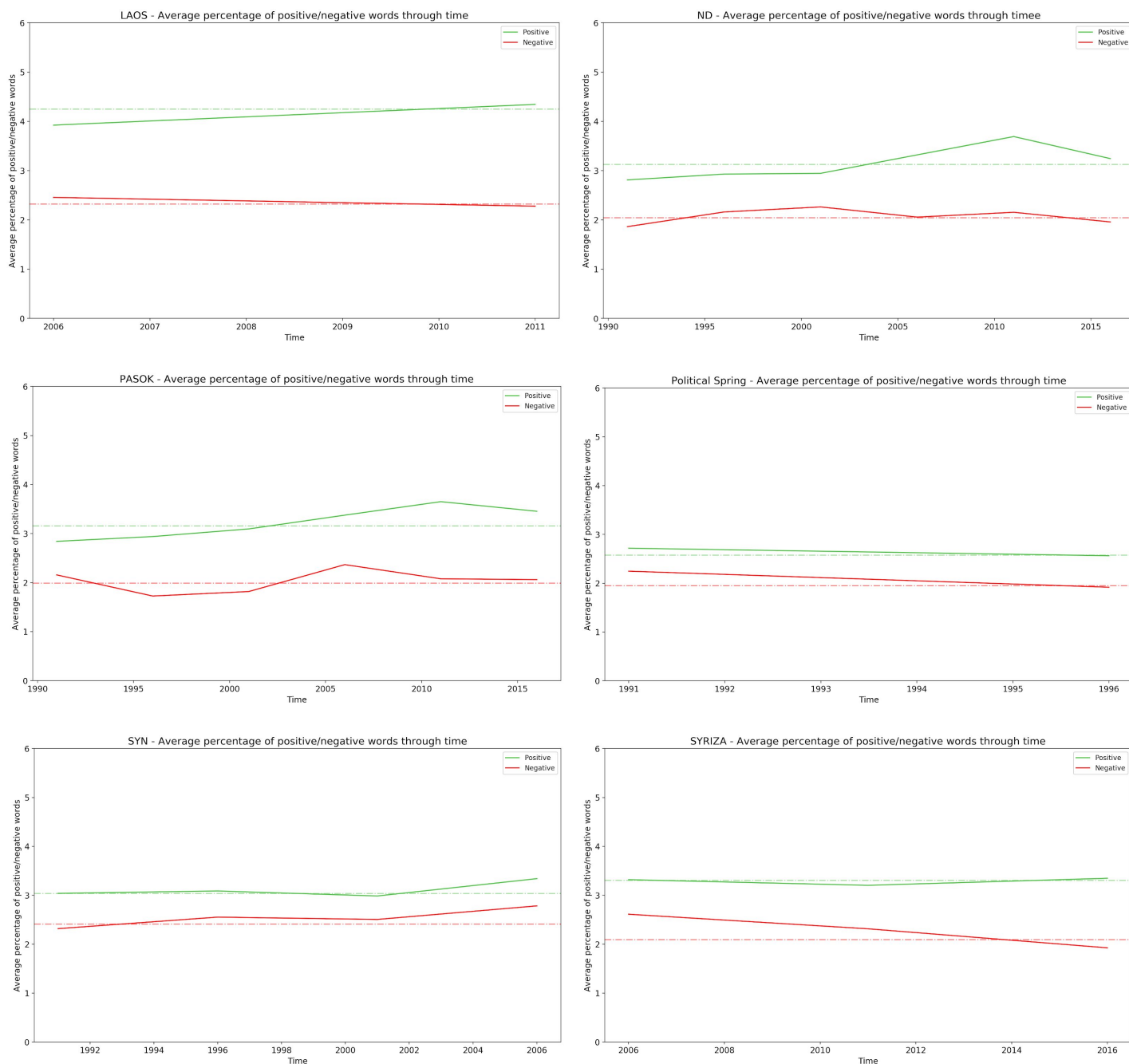


Figure 21: Average percentage of positive and negative words for each political party from 1989 to 2017



Furthermore, we present two additional graphs which showcase the percentage of positive and negative words for each party, cumulatively for all the years from 1989 until 2017.

Concerning the positive words, we can see that DIMAR holds the higher score with a notable difference from the rest of the political parties. On the other hand the parties of Political Spring and Alternative Ecologists score with the lowest percentage of positive words, though in close proximity with the rest of the parties. This graph shows greater variations in the percentages for each political party, with values ranging between 2.5 and 5.0. This comes in contrast with the graph for negative words, where the range of values is smaller, as you can see below.

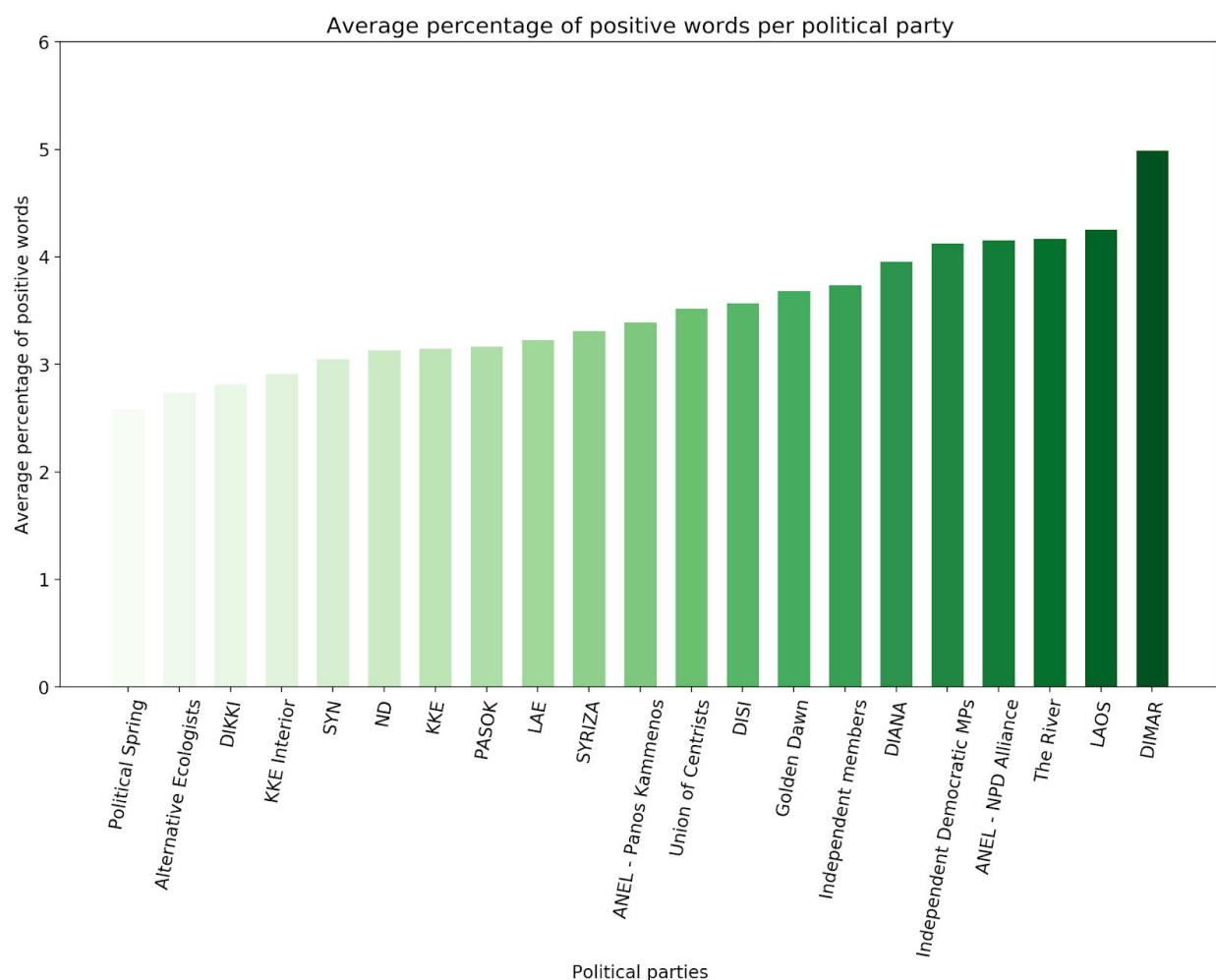


Figure 22: Average percentage of positive words for each political party during all years from 1989 until 2017

Concerning the negative words, the parties with the higher percentage of negative words at all times are DIANA and the “Alternative Ecologists”. The parties with the lowest percentage of negative words are the “Union of Centrists” and “ANEL - National Patriotic Democratic Alliance”. All values are in close proximity, expanding between 1.5 and 2.5.

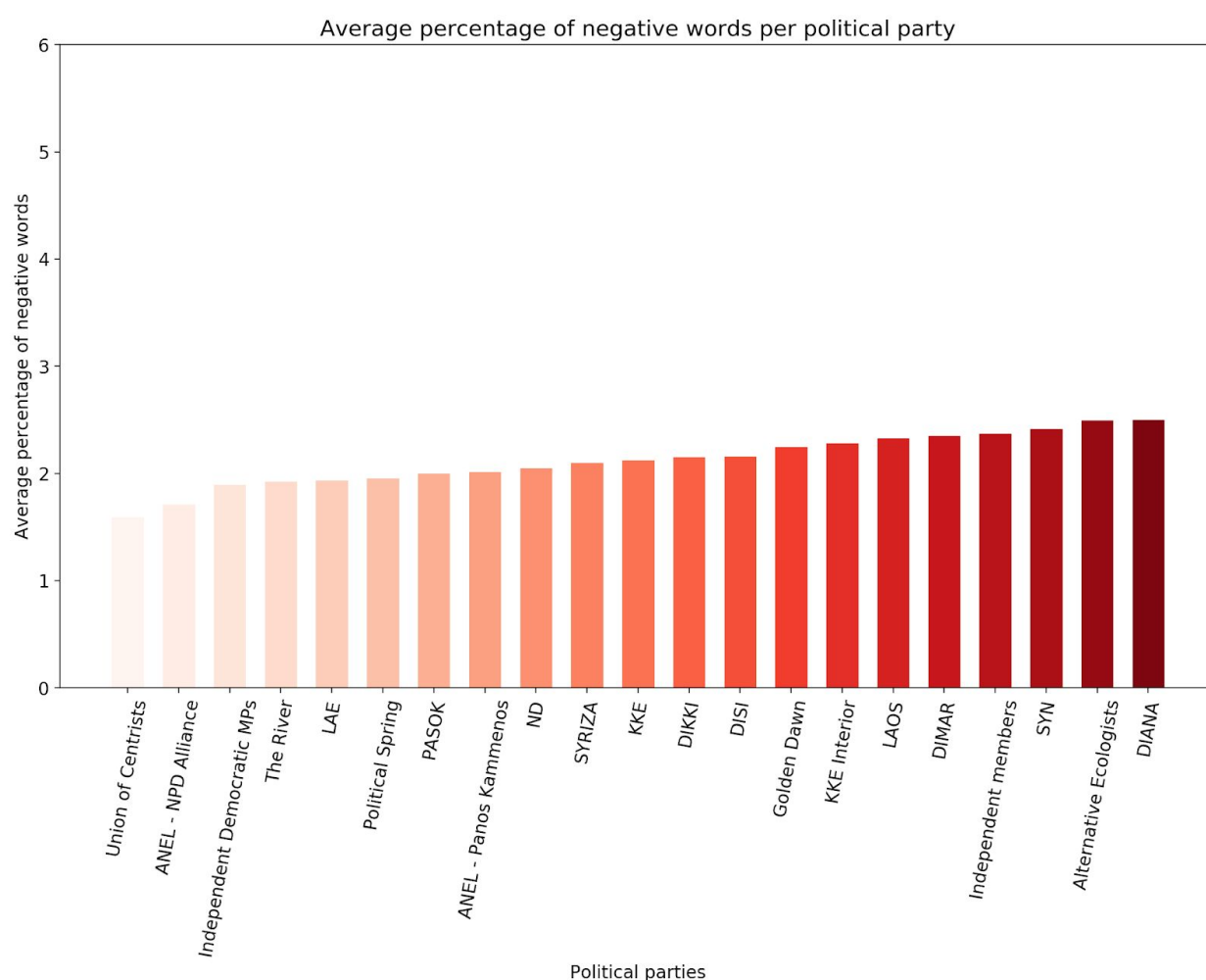


Figure 23: Average percentage of negative words for each political party during all times from 1989 until 2017

6. Conclusion

At this project, we scraped from the Hellenic Parliament website 4905 record files of the parliament sittings from 1989 up to 2017. We also collected information of all the parliament members from the “Restoration of Democracy” (Greek polity change) in 1974 up to today. We preprocessed our data and matched each speech in the records with the corresponding official name of the parliament member. Then, we evaluated our dataset in regard to the readability of the speeches and we performed lexicon-based sentiment analysis.

Concerning the readability evaluation, we measured the readability and speech quality with the use of the SMOG Index formula. We created our own SMOG Index calculator for the Greek language. However, as the application of SMOG index on the Greek language lacks statistical validity, we interpreted the results comparatively, without matching them to prerequisite years of schooling. Our findings indicate an important drop on the average readability score of the parliament sitting records from 2003 up to 2017. The lowest speech quality is held by the political parties “Union of Centrists”, “Golden Dawn” and “ANEL - National Patriotic Democratic Alliance” (not to be confused with ANEL - Panos Kammenos political party). The highest speech quality is held by the political parties “Democratic Social Movement” (DIKKI), “Alternative Ecologists” and the “Coalition of the Left, of Movements and Ecology” (SYN).

Concerning the sentiment analysis, we used two different Greek sentiment lexicons and evaluated the presence of a range of different sentiments. Specifically, the Lexicon 1 provided ratings for the sentiments of anger, disgust, fear, happiness, sadness and surprise. Lexicon 2 provided a list of positive and negative words. The sentiment ratings of our dataset are a result of direct calculations derived from the words constructing the speeches. The results on Lexicon 1 suggest that the communication among parliament members throughout all times is characterized mainly by the feeling of surprise followed closely by anger and disgust. The less common sentiment encountered is sadness. Happiness and fear fall in between the most common sentiments and sadness. The score of the sentiments is quite steady throughout the years. The results on Lexicon 2 show a steady prevalence of positive over negative words throughout the years examined.



The main difficulties that we coped with during this project were the anarchic structure and the many mistakes in the parliament records as well as the very few tools available for readability and sentiment analysis on the Greek language.

7. Future work

There is so much more to be done with the dataset of the Hellenic Parliament records. One important first step would be to adjust the SMOG Index formula to the Greek language and match the resulting scores with the years of Greek schooling needed in order for someone to understand a given Greek text.

Concerning the field of sentiment analysis, we could utilize linguistic data such as part of speech tagging and, furthermore, implement hate speech detection and toxicity detection with the use of neural networks and deep learning.

We could also mine the dataset in search of correlations such as that of the member's age or gender with the key subjects of their speech and their views on specific topics. Other interesting findings can include the identification of the topics that are discussed every year and the evolution of the discussions on each topic over the years.

Source code availability

The source code of the project is available on Bitbucket in the following link <https://bitbucket.org/kdritsa/greekparliament/>



8. References

- [1] S. Rosenthal, N. Farra & P. Nakov, “SemEval-2017 Task 4: Sentiment Analysis in Twitter”, in *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*, *Computational Linguistics*, Vancouver (Canada), August 2017, pp. 502-518; <http://www.aclweb.org/anthology/S17-2088>
- [2] D. Mallis, G. Kalamatianos, D. Nikolaras & S. Symeonidis, “Sentiment Analysis of Greek Tweets and Hashtags using Sentiment Lexicon”, New York (USA), *ACM*, 63–68. doi:10.1145/2801948.2802010; <http://hashtag.nonrelevant.net/Sentiment%20Analysis%20of%20Greek%20Tweets%20and%20Hashtags%20using%20Sentiment%20Lexicon.pdf>
- [3] M. Taboada, J. Brooke, M. Tofiloski, K. Voll & M. Stede, “Lexicon-Based Methods for Sentiment Analysis”, *Computational Linguistics*, Vol 37 - Issue 2, June 2011, p.267-307; https://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00049
- [4] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, E. Herrera-Viedma, “Sentiment Analysis: A Review and Comparative Analysis of Web Services”, *Information Sciences*, Vol. 311, August 2015, pp. 18-38
- [5] M. Ghiassi, J. Skinner & D. Zimbra, “Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network”, *Expert Systems with Applications*, Vol. 40 - Issue 16, 2013, pp. 6266–6282; 2013; http://kt.ijs.si/markodebeljak/Lectures/Seminar_MPS/2012_on/Seminars_2015_16/Simon%20Brmez/Bibliography/%5B20%5D%20Twitter%20brand%20sentiment%20analysis%20A%20hybrid%20system%20using%20n-gram%20analysis%20and%20dynamic%20artificial%20neural%20network.pdf
- [6] W. Medhat, A. Hassan & H. Korashy, “Sentiment analysis algorithms and applications: A survey”, *Ain Shams Engineering Journal*, Vol. 5 - Issue 4, December 2014, pp. 1093-1113 ; https://ac.els-cdn.com/S2090447914000550/1-s2.0-S2090447914000550-main.pdf?_tid=5d3



[432f9-6be8-4e78-806b-0703527d2dc1&acdnat=1525129896_c7f354211261c679ccadc8e8a4656b30](https://doi.org/10.1007/978-1-4939-9999-9_432f9-6be8-4e78-806b-0703527d2dc1&acdnat=1525129896_c7f354211261c679ccadc8e8a4656b30)

[7] Mikalai Tsytsarau, Themis Palpanas, “Survey on mining subjective data on the web”, *Data Mining and Knowledge Discovery*, Vol. 24 - Issue 3, pp 478–514, May 2012; <https://link.springer.com/article/10.1007%2Fs10618-011-0238-6>

[8] M. Karamibekr & A. A. Ghorbani, "Sentence Subjectivity Analysis in Social Domains," *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Atlanta, GA, 2013, pp. 268-275, doi: 10.1109/WI-IAT.2013.39;

https://www.researchgate.net/publication/262163060_Sentence_Subjectivity_Analysis_in_Social_Domains

[9] H. Binali, V. Potdar & C. Wu, "A state of the art opinion mining and its application domains" *IEEE International Conference on Industrial Technology*, Australia, 2009, pp. 1-6. doi: 10.1109/ICIT.2009.4939640

[10] B. Liu, “Sentiment Analysis and Opinion Mining”, *Synthesis Lectures on Human Language Technologies*, May 2012, pp. 1-167, Morgan and Claypool Publishers, 2012.

[11] M. Gamon, A. Aue, S. Corston-Oliver & E. Ringger, “Pulse: Mining customer opinions from free text,” in *Proceedings of the 6th International Symposium on Intelligent Data Analysis*, 2005, pp. 121–132.

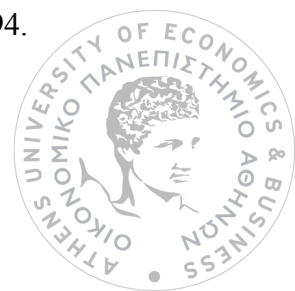
[12] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery: opinion extraction and semantic classification of product reviews,” in *Proceedings of the 12th international conference on World Wide Web, ACM*, 2003, pp. 519–528.

[13] P. D. Turney, “Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews,” in *Proceedings of the 40th Annual Meeting, Association for Computational Linguistics*, 2002, pp. 417-424.

[14] T. Nasukawa & J. Yi, “Sentiment analysis: capturing favorability using natural language processing,” in *Proceedings of the 2nd international conference on Knowledge Capture, ACM*, 2003, pp. 70-77.



- [15] S. Somasundaran & J. Wiebe, “Recognizing stances in ideological on-line debates,” in *Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, ACM, 2010, pp. 116-124.
- [16] X. Ding, B. Liu & L. Zhang, “Entity discovery and assignment for opinion mining applications,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 1125-1134.
- [17] B. Pang, L. Lee & S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, July 2002, pp. 79-86; <http://www.aclweb.org/anthology/W02-1011>
- [18] M. Mohri, A. Rostamizadeh & A. Talwalkar, “Foundations of Machine Learning”, MIT Press, 2012; [doc.nit.ac.ir/cee/jazayeri/MachineLearning/Mehryar_Mohr%202012/\[Mehryar_Mohri_Afshin_Rostamizadeh_Ameet_Talwalkar\(BookFi.org\).pdf](http://doc.nit.ac.ir/cee/jazayeri/MachineLearning/Mehryar_Mohr%202012/[Mehryar_Mohri_Afshin_Rostamizadeh_Ameet_Talwalkar(BookFi.org).pdf)
- [19] A. Abbasi, H. Chen & A. Salem, “Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums”, *ACM Transactions on Information Systems*, Vol. 26 - Issue 3, June 2008, Article No. 12; <https://dl.acm.org/citation.cfm?doid=1361684.1361685>
- [20] M. Gamon, “Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis”, in *Proceedings of International Conference on Computational Linguistics*, 2004, p. 84.
- [21] E. Riloff, J. Wiebe & T. Wilson, “Learning subjective nouns using extraction pattern bootstrapping”, in *Proceedings of the 7th Conference on Natural Language Learning*, Canada, 2003, pp. 25–32.
- [22] M. Efron, “Cultural orientations: Classifying subjective documents by cocitation analysis”, in *Proceedings of the AAAI Fall Symposium Series on Style and Meaning in Language, Art, Music, and Design*, 2204, pp. 41-48.
- [23] C. Johnston, F. P. Rivara & R. Soderberg, “Children in Car Crashes: Analysis of Data for Injury and Use of Restraints”, *Pediatrics*, Vol. 93 - Issue 6, pp. 960-965, June, 1994.



- [24] M. Wegner & D. Girasek, “How readable are child safety seat installation instructions?” *Pediatrics*, Vol. 111 - Issue 3, pp. 588-591, March 2003.
- [25] W. H. DuBay, “The Principles of Readability”, *Impact Information*, August 2004.
- [26] G. R. Klare, “The measurement of readability”, *Iowa State University Press*, 1963;
<https://babel.hathitrust.org/cgi/pt?id=mdp.39015004229558;view=1up;seq=7>
- [27] G. H. McLaughlin, “SMOG grading - a new readability formula”, *Journal of Reading*, Vol. 12 - iSSUE 8, May, 1969, pp. 639-646.
- [28] E. Dale & J. S. Chal, “The concept of readability”, *Elementary English*, Vol. 26 - Issue 1, January 1949, pp. 19-26.
- [29] R. Flesch, “A new readability yardstick”, *Journal of Applied Psychology*, Vol 32 - Issue 3, June 1948, pp. 221-233.
- [30] J. N. Farr, J. J. Jenkins & D. G. Paterson, “Simplification of the Flesch Reading Ease Formula”, *Journal of Applied Psychology*, Vol. 35 - Issue 5, October 1951, pp. 333-357.
- [31] R. Gunning, “The technique of clear writing”, *New York: McGraw-Hill*, 1952.
- [32] J. Seely, “ “Chapter 10: Audience” - Oxford Guide to Effective Writing and Speaking: How to Communicate Clearly”, *Oxford University Press*, 2013, pp. 120-123;
https://books.google.gr/books?id=hVEGAQAAQBAJ&pg=PA1&redir_esc=y#v=onepage&q&f=false
- [33] A. Contreras, R. Garcia-Alonso, M. Echenique & F. Daye-Contreras, “The SOL Formulas for Converting SMOG Readability Scores Between Health Education Materials Written in Spanish, English, and French”, *Journal of Health Communication*, Vol. 4 - Issue 1, 1999, pp. 21-29.
- [34] W. L. Taylor, “ “Cloze Procedure”: A new tool for measuring readability” ”, *Journalism & Mass Communication Quarterly*, Vol. 30 - Issue 4, 1953, pp. 415-433.
- [35] D. Tzimokas & M. Mattheoudaki, “Δείκτες αναγνωσιμότητας: Ζητήματα εφαρμογής και αξιοπιστίας” in “Major Trends in Theoretical and Applied Linguistics”, Vol. 3, 2014, pp. 367-384.



- [36] A. Tsakalidis, “Greek Sentiment Lexicon”. Available online: <https://github.com/MKLab-ITI/greek-sentiment-lexicon>
- [37] “Multilingualsentiment”, Data Science Lab, Stony Brook University, New York. Available online: <https://sites.google.com/site/datasciencelab/projects/multilingualsentiment>
- [38] P. Louridas, “invoices”. Available online: <https://github.com/louridas/invoices>
- [39] M. A. Jaro, “Advances in record linkage methodology as applied to the 1985 census of Tampa Florida”, *Journal of the American Statistical Association*, Vol. 84 - Issue 406, June 1989, pp. 414-420.
- [40] W. E. Winkler, “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage”, *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 1990, pp. 354-359.
- [41] “pygtrie” python library. Available online: <https://github.com/google/pygtrie>
- [42] “Apache Tika - a content analysis toolkit”, Available online: <https://tika.apache.org/>
- [43] M. N. Tod, “Three Greek Numeral Systems”, *The Journal of Hellenic Studies*, 1913, Vol. 33, pp. 27-34.
- [44] Y. He and D. Zhou, “Self-training from labeled features for sentiment analysis”, *Information Processing & Management*, Vol. 47 -Issue 4, July 2011, pp. 606-616.
- [45] F. Xianghua, L. Guo, G. Yanyan and W. Zhiqiang, “Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon”, *Knowledge-Based Systems*, Vol. 37, January 2013, pp. 186-195.
- [46] K. Kim and J. Lee, “Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction”, *Pattern Recognition*, Vol. 47 - Issue 2, February 2014, pp. 758-768.
- [47] A. C. König and E. Brill, “Reducing the human overhead in text categorization”, in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '06*, ACM Press, 2006, pp. 598-603.
- [48] Encoding sets supported by Java SE 8; Available online: <https://docs.oracle.com/javase/8/docs/technotes/guides/intl/encoding.doc.html>

