# Final Report

Bachelor’s thesis - Describing the language of the Dutch House of Representatives

Rebecca Glans, Mila Hendrikse, Martin Koole

July 16, 2017

# Preface

For our Bachelor's Thesis at the Technical University of Delft (TU Delft) we performed a project with the goal of exploring possible descriptions of the language of politicians of the Dutch House of Representatives. Our client is dr. Cynthia Liem [1], Assistant Professor at the Multimedia Computing Group at the TU Delft and our project coach is dr. ir. Alessandro Bozzon[2], Assistant Professor with the Web Information Systems group at the TU Delft.

[1] http://mmc.tudelft.nl/users/cynthia-liem

[2] http://www.wis.ewi.tudelft.nl/bozzon/

## Acknowledgements

We would like to thank our supervisor, dr. ig. Alessandro Bozzon for guiding us through both the research and development process. His often straight forward feedback allowed us to make decisions which directed our Bachelor's thesis in the right direction when we were uncertain of what to do, and showed us the possibilities we could consider. We would also like to thank our client, dr. Cynthia Liem, for being highly involved during each phase of the Bachelor's thesis. From Dr. Liem we received many suggestions and remarks on our product and report, helping us to improve them every time. Thank you both for showing so much enthusiasm during the span of this Bachelor's thesis, being willing to meet with us weekly, and keeping up via chats regularly. We are also very grateful for the continuous help of Otto Visser, assistant professor at the Distributed Systems Group, who - even though he was not an official supervisor - always answered our questions and was ready to help throughout the thesis. Otto Visser was also present at our midterm presentation as well as our final presentation. A special thanks to prof. dr. Rinie Geenen[3] of the University of Utrecht for granting us access to the Dutch translation of the Linguistic Inquiry and Word Count (LIWC) dictionary.

[3] https://www.uu.nl/medewerkers/MJMGeenen/0

# Summary

Political content reaches civilians via social media more and more nowadays. This content is often biased and filtered depending on the user's connections on social media. However, almost nobody watches actual debates or reads reports from the actual Dutch House of Representatives (DHoR), where the most influential decisions are made. With our Bachelor's thesis we aim to give users an insight into the language used by Representatives in the DHoR.

This Bachelor thesis can be divided into two parts, the first part is the research phase, the second part the development of the Data Analysis Tool. During the research phase we searched for ways that would allow us to describe the use of language in the DHoR, and the resources necessary to perform corresponding calculations. We defined several non-topical text features to perform the description: complexity, sentiment analysis and the rate of femininity / masculinity. For each one of these features, we will calculate their respective values based on the particular language of a Representative. Besides these non-topical text features, we also introduce a topical text feature. As opposed to the first, topical text features actually focus on the content of a text, rather than the language used. Some politicians are known to act differently regarding certain circumstances. By introducing filters, we can explore the effects or influences these circumstances have on the use of language (text features) of Representatives. Finally, we examined the available data from the DHoR and how to access it.

The second part of this thesis is the development of our Data Analysis Tool (DAT). In order to calculate scores for the text features, we used reported data from debates and discussions in the DHoR from the past five years. We developed a DAT that can be used to explore the use of language of Representatives by:

- analysing individual Representative text feature scores
- comparing the average text feature scores of all Representatives
- allowing to plot the text feature scores of Representatives over time and filter on time

## Structure of report

In this document, you will find an introduction, which explains the purpose of this project, followed by a problem definition. Then we will define the research questions; main and sub-questions. In the next chapter, the data of the DHoR will be discussed, which is proceeded by a chapter describing how we are going to represent the language of the DHoR; text features. Then, the research part of the thesis will be concluded by the requirement analysis.

In chapter seven, the one that marks beginning of the second part of the thesis, we describe the development of the DAT. Then we will discuss our results, followed by the discussion. Thereafter, we reflect on the research and development process, and finally, a chapter that contains our conclusion(s) and a section discussing possible future work.

# Contents

# 1 Introduction

The purpose of this project is to analyse textual data from the Dutch House of Representatives (DHoR) in order to describe the language of Representatives and visualise it. The project was inspired by the current social media filter bubble. This bubble refers to the phenomenon where online social media users get separated from online content that opposes the users' viewpoints. This isolation is due to and strengthened by personal online behaviour (clicks, likes, posts, search history and more) in combination with personalised website filtering algorithms. As this phenomenon expands over time, it enhances the political polarisation within a society; people tend to refuse being positive about politicians they (completely) disagree with. In this online "bubble" a user is only exposed to a tiny part of all information about politicians' remarks, decisions and activities and derives his/her opinions and conclusions from this biased and limited information. It is questionable if the root of these isolated groups of political followers lies purely with their social and online setting which offers them limited information, or if the language used by different politicians has such a great influence on followers that they could never remotely appeal to certain groups of civilians (for more detailed information on the social media and its effects, please have a look at Appendix B). Therefore we will not look at limited and biased content, but to the actual words spoken in actual debates and discussions in the DHoR. We will examine (computable) textual aspects (text features) which could play a part in the impression Representatives can leave on their listeners. We aim to give users an insight into the language used in the DHoR and the ability to draw their own conclusions based on the visualisation(s) of text features, as this application uses completely unbiased, raw data, ensuring the absence of filter bubbles.

# 2 Problem definition and analysis

In order to develop a Data Analysis Tool to describe and visualise language, a framework for the description of human language is needed. How can something as complex as human language be simplified to abstract values? The results of this framework should be shown with clear data visualisation.

For the Data Analysis Tool textual data of reported debates and discussions from the Dutch House of Representatives needs to be retrieved. Then a pipeline must be developed to access, process and store this data. At last a User Interface (UI) needs to be developed to allow a user to interact with the processed data to analyse the language in the Dutch House of Representatives.

# 3 Research question

We use the text-features described in Section 5 to visualize the language used during debates and discussions in the Dutch House of Representatives.

## Main question

How can we describe the language in the Dutch House of Representatives?

## Sub questions

1. How is a Representative's language characterised by text features?
2. How is a Representative's language (based on the text features) influenced by:
   - the topic discussed
   - major (planned) events on the political agenda
   - unforeseen events of political importance
3. What clusters of Representatives can we distinguish when visualizing their speech patterns?

# 4 Data of the House of Representatives

Every spoken word during an official meeting in the Dutch House of Representatives is documented by the ”Dienst Verslag en Redactie” and published online. These so-called “shorthand” reports (documented speech) tell us everything that was discussed and which members participated. Meetings accur in two ways: plenary or in committees. The former is the one we are most familiar with, as it takes place in the plenary hall. In smaller rooms, Representatives meet in committees. Every Representative takes part in at least one committee. Among the over 30 committees are for instance the ”Committee for Education, Culture and Science” and ”Committee for Finance”[4].

## 4.1 Data source

Publication of the reports happens in two stages. Almost immediately after a meeting has taken place, the unofficial version is published on the House's website. Whereas a plenary meeting is completely documented [5], a committee meeting is only documented during the activities stated below[6]. Explanation on how the data will be retrieved can be found in section 7.1.

## 4.2 Description of available data

Different activities occur during meetings in which we can analyse the politicians' language. Below, these are listed with a short explanation.

### Committee meetings

The different forms of discussion during committee meetings are:

- General Discussion (Algemeen Overleg)
- Discussion on legislation (Wetgevingsoverleg)
- Dicussion on policy papers (Nota-overleg)

The general discussion is the most common. The committee views a specific aspect of a minister's or state secretary's policy, who is present as well. The committee asks various questions, which the member of cabinet answers. Concluding this discussion or submitting motions regarding the subject is done during plenary meetings as decisions must be made by the whole House of Representatives. The other two forms of discussion are used to discuss respectively bills or policy papers with the involved member of cabinet.

[4] https://www.tweedekamer.nl/kamerleden_en_commissies/commissies

[5] https://www.tweedekamer.nl/kamerstukken/plenaire_verslagen

[6] https://www.tweedekamer.nl/kamerstukken/verslagen

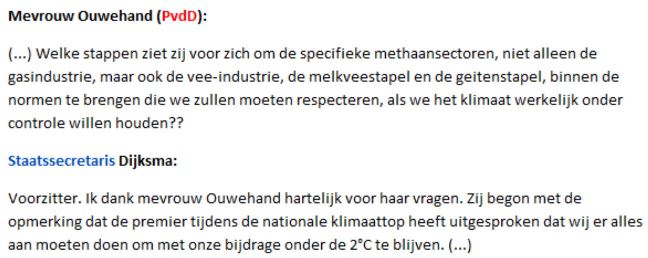
### Plenary meetings

The most frequent plenary activities are [7]:

- Questions round (Vragenuur)
- Adjusting the current agenda (Regeling van Werkzaamheden)
- Plenary Debate
- Voting

Politicians can submit questions to the chairman of the plenary meetings. These questions can be addressed to any member of the cabinet, meaning they are always addressed to the ministers or state secretaries. Like the discussions with the committees, it is one of the primary ways used to control our government[8]. The chairman can suggest adding items to the agenda and Representatives can agree or disagree. For the debates Representatives deliver various motions (often as result of a committee meetings) and the House reacts (debates). After a debate there is a voting to make the DHoR's final decision.

Figure 1: Text sample of a questions round



**Mevrouw Ouwehand (PvdD):**

(...) Welke stappen ziet zij voor zich om de specifieke methaansectoren, niet alleen de gasindustrie, maar ook de vee-industrie, de melkveestapel en de geitenstapel, binnen de normen te brengen die we zullen moeten respecteren, als we het klimaat werkelijk onder controle willen houden??

**Staatssecretaris Dijksma:**

Voorzitter. Ik dank mevrouw Ouwehand hartelijk voor haar vragen. Zij begon met de opmerking dat de premier tijdens de nationale klimaattop heeft uitgesproken dat wij er alles aan moeten doen om met onze bijdrage onder de 2°C te blijven. (...)

### Topics

For determining the topics which will cover the overall data, we will use the subjects the committees address and the parties' agenda (and visions) at the time of entering the DHoR. The parties' data is (naturally) public as well and can be found on their websites.

[7] https://www.tweedekamer.nl/debat_en_vergadering/plenaire_vergaderingen
[8] https://nl.wikipedia.org/wiki/Kamervraag

# 5 Representing the language of the DHoR

In order to analyse the way Representatives speak, we examine (computable) textual aspects which could play a part in the impression politicians can leave on their listeners. Not only the content of a conversation, a discussion or (political) proposal influences this impression. Also choice of words and sentences of a speaker influences the way a message is transmitted and perceived[1]. We call these textual aspects 'text features', and we divide them into two groups: non-topical and topical text features. We aim to describe the language used in the House of Representatives via these text features. We will initially not look at what is said , but how it is said textually. As shown in Section 5.1 this will include, among others, a sentiment analysis and an analysis of the complexity of the speech. The topic of a speech is needed for our analysis as well, meaning we will look at what is said at a high-level. With these defined text features which describe speech (in text form), we can analyse and measure how speech of Representatives varies among each other or under different circumstances. In Section 5.3 we introduce three filters with which we want to visualize influences by circumstances.

## 5.1 Non-topical text features

Intelligent use of language can help politicians make their statements clear, make listeners memorize what is said easier, persuade their colleagues into considering a topic from a different perspective, emphasize their powerful position and much more. We try to turn these influential linguistic aspects into numbers with the text features.

Also we will calculate the average of feature values per politician, which we call the baseline of the politician.

In this section we describe all the text features to be used and explain their relevance. In section 7.2.2 we show how they are computed.

- Complexity
- Sentiment Analysis
- Synonyms (a supporting feature)
- Feminine to masculine speech style ratio

### Complexity

How difficult is it to understand a politician? Does he use simple (short) words or words containing many syllables? It can indicate what the speaker expects his audience to know about the topic or, on the other hand, how well-informed the speaker is. Using the Flesch-Kincaid Reading Test [2] we can determine the difficulty of a given text based on the amount of syllables in each word.

### Sentiment Analysis

Different types of texts can be categorized in two main sentimental categories, namely positive and negative [3]. Of course, it is possible to assign a value that lies somewhere between these; defining a scale that allows intermediate values. This can thus also be done on mere word or sentence level [4]. In general this means it is possible to indicate per (piece of) text whether it is linked to positive or negative emotional value. Furthermore, sentiment analysis can be used to analyse political debates[5].

### Synonyms (a supporting feature)

Analysing which words are used can also give an insight in how politicians (wish to) communicate. When engaging in a topic, does one choose a more **modern** or **traditional** version of a word? Does one introduce a self made word to emphasize his opinion or use slang to add humour or show involvement? This feature is also contained in the features above, which is why it is also a supporting feature. For instance, in [6] it is shown that synonyms can carry **sentiment** in certain contexts; instead of using the word abortion, medical terms are used when in a pro-choice position.

### Feminine to masculine speech style ratio

Politics is mostly a male-dominated area. While women are entering decision-making bodies more and more each year, they still form a minority. Because of this, they tend to conform to the style of communication typical for this area that, as a result of the majority of the members being male, has a tendency towards a masculine style of speech [7]. What is interesting about this text feature is that the gender indicating speech of a politician (or human in general) is not fixed; it can deliberately or even unintentionally fluctuate over time or even between different speeches and debates for the same politician. Take for example United States politician Hilary Clinton. A research about the linguistic style of Hilary Clinton on data from interviews and candidate debates dating between 1992 and 2013 shows that Clinton's linguistic style fluctuates between masculinity and femininity over the years, depending on the different roles she played over the years (some years it was a more supportive role, others one of pure leadership) [8].

The Netherlands scores really well on the global Gender Inequality Index (GII)[9], ranking 3rd most equal country in 2015 according to the United Nations Development Programme[9] while the USA for example ranked 43. The Dutch parliament also scored high globally on percentage of women in parliament on the 1st of March 2017 (so the ratio of the parliament before the most recent elections) with a score of 34.7% ranking 21st place in a report made by the Inter-Parliamentary Union [10]. Because of this relatively powerful ranking of the Netherlands compared to other countries we include a text feature about

[9] http://hdr.undp.org/en/content/gender-inequality-index-gii

gender indicative speech. We believe it is interesting how feminine/masculine our House of Representatives speaks and if this speech changes over time and to what extend it differs between female and male politicians.

The ratio of the feminine to masculine speech style can be easily calculated by counting feminine and masculine linguistic markers in text and dividing them: feminine/masculine[8].

## 5.2 Topical Text features

As said before, we will not be looking at what is actually said during a speech. However, in the case of politicians, the topic of his/ her speech can be interesting. To this extent we will look at the content of a speech.

### Topics

The topic of a text is in fact a text feature. We will use it to describe which topics a Representative addresses and in what amount. It gives insight in how much the Representative's party or committee influences the content of his/ her speech.

## 5.3 Filters

Can we analyse how the speech pattern of a politician varies during his/her term? Many things could influence how a politicians speaks; experience, the topic discussed, recent events. Next to giving an overall speech pattern, we would like to show how this speech pattern was formed or influenced. Our first approach is using the topic discussed (5.2). In this section we would like to give two approaches that use different events or periods during a politician's term to determine their influence.

### Topics

During election campaigns each party announces which current issues it wants to address and how it wants to tackle them. These issues can vary from economical to immigration topics and many more. When a party makes it into the government it is expected of them to take action and work on the issues that were campaigned on in order to satisfy its voters and gain trust. We want to see if these specific party topics, because they are of greater political importance to the party than others, have an influence on the speech of party members compared to topics that weren't mentioned in the party campaign.

### The political agenda

The date of a debate could change the politicians speech pattern; some politicians for example are known for making more populist statements when the elections are approaching. By taking different (major) events on the political agenda we can analyse how the speech patterns vary.

### Unforeseen important events

The attitude and speech pattern of politicians can vary when news emerges that causes immediate response from- and debates in the House of Representatives. By looking at so-called emergency debates in the House of Representatives we can analyse how the values of the text features of politicians vary under circumstances that urges them to think and debate with little preparation time about stressing topics.

# 6 Requirements analysis

In this chapter we will describe what kind of product we aim to obtain through implementation, the MoSCoW analysis method that indicates the priority of each of our requirements, a short paragraph on scope and limitations, and our initial planning for the span of the project.

## 6.1 Implementation goal

With our product we want to grant a different insight into the House of Representatives to voters. In our web-application we want to create a dynamic overview that already gives some details. However, the user is free to select and examine different politicians and topics or alter filters in order to interactively find overviews that are interesting to him/her. By clicking around and filtering, the user will explore expected and unexpected relatedness and unrelatedness between topics, text features, political parties and politicians.

## 6.2 Product Vision

In figure 2 we can see the the so-called landing page. Our website can be seen as a dashboard for exploring the House of Representatives. The mid section is initially filled with brief information about our project and how to use this website. On the right side the menu for selecting a politician will already be open. When selecting a politician, the general information will be removed and the mid section will change accordingly.

In figure 3 the page is shown after a user has clicked on the politician he or she wants to know more details about. The mid section now shows the topics (the circle-shaped labels in the midst of the screen) that are of great relevance to this particular politician. In the column to the left, we show a portrait of the politician, and underneath it, we include some personal information, for example the baseline, and some more insight of the party he or she belongs to. In the right column, we show a list of actuality (important recent incidents), his or her corresponding party topics, and important dates (milestones) to this particular politician.

Once a topic has been chosen, the user will now see the page as in figure 4 the mid section will show the politician's position relative to his house members. The user can highlight the different politicians (dots) according to party or can filter them on different ratings for the text features.

Figure 2: Landing page

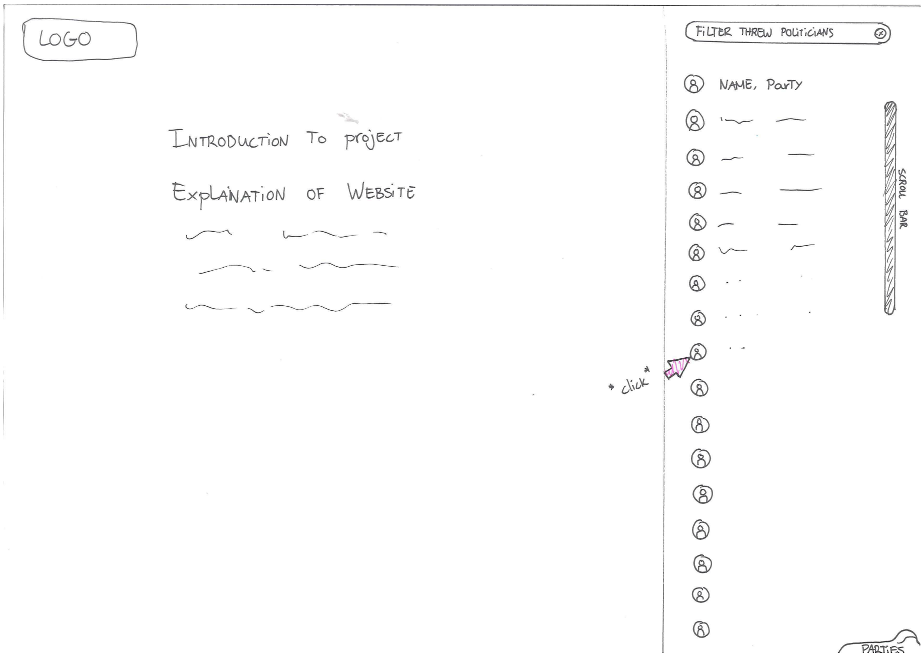


Figure 4: When clicked on debate topic: dashboards

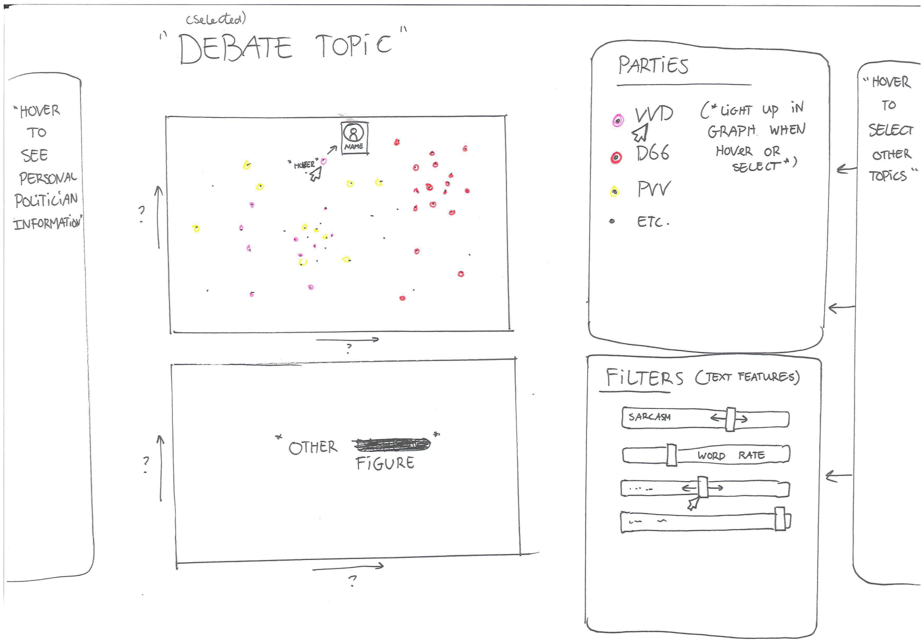
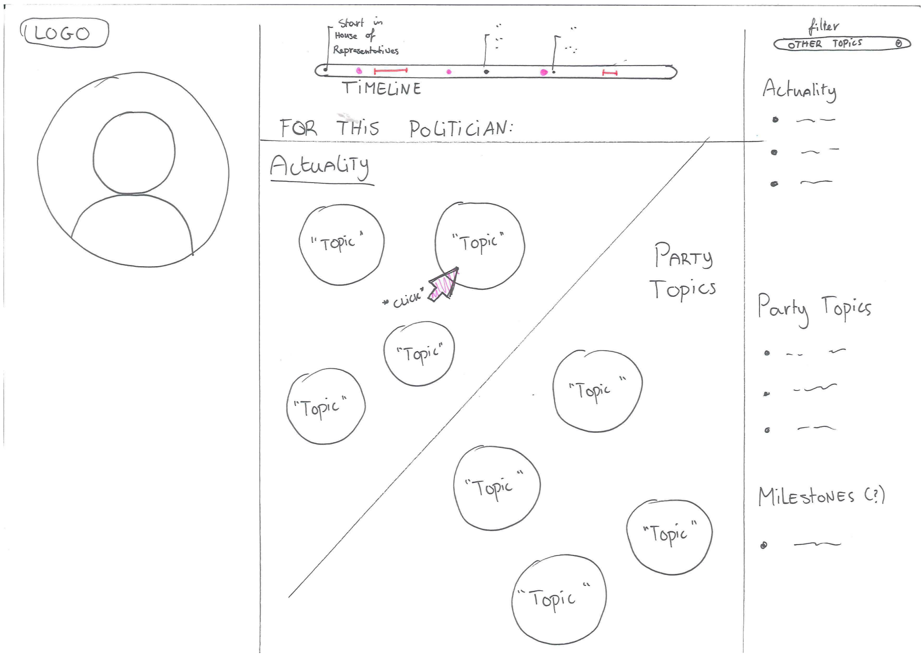

(selected)
"DEBATE TOPIC"
"HOVER TO SEE PERSONAL POLITICIAN INFORMATION"
"HOVER"
NAME
?
?
"OTHER FIGURE"
?
?
PARTIES
VVD
D66
PVV
ETC.
(*LIGHT UP IN GRAPH WHEN HOVER OR SELECT*)
"HOVER TO SELECT OTHER TOPICS"
FILTERS (TEXT FEATURES)
SARCASM
WORD RATE

Figure 3: When clicked on politician: topic selection



## 6.3 MoSCoW analysis

In order to manage our priorities we created MoSCoW tables. In Tables 5 and 6 you will find our requirements, prioritized using the MoSCoW method, for our data analysis and web-application respectively.

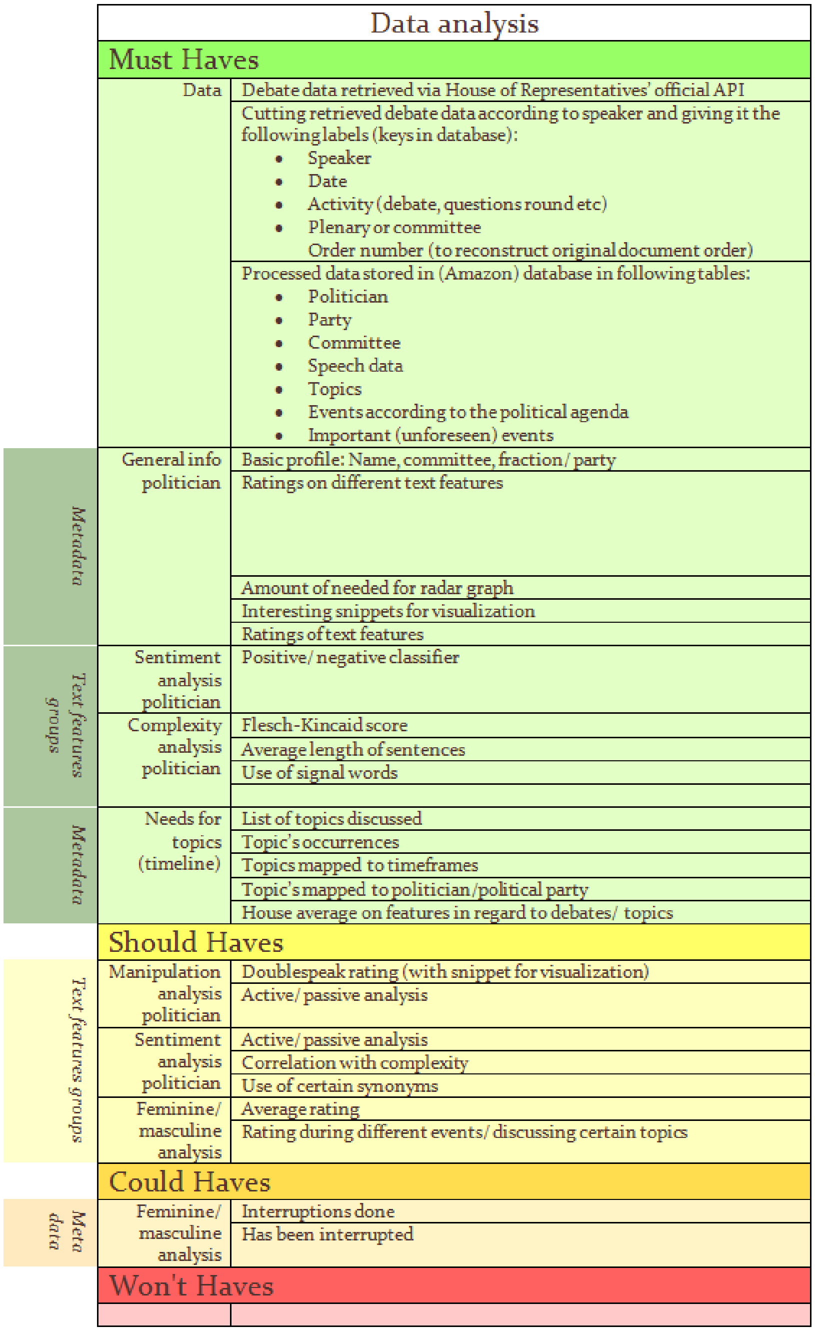
## 6.4 Scope and limitations

The product of our Bachelor's thesis will be a framework for language analysis in the DHoR, including a web-application. We have a total amount of 9 weeks to complete the project. Not only do the literal words of a politician influence our opinion on the person, but also other aspects like for example; body language, tone of speech and facial expressions. Since we are provided with a limited amount of time, we will only focus on text.

## 6.5 Planning

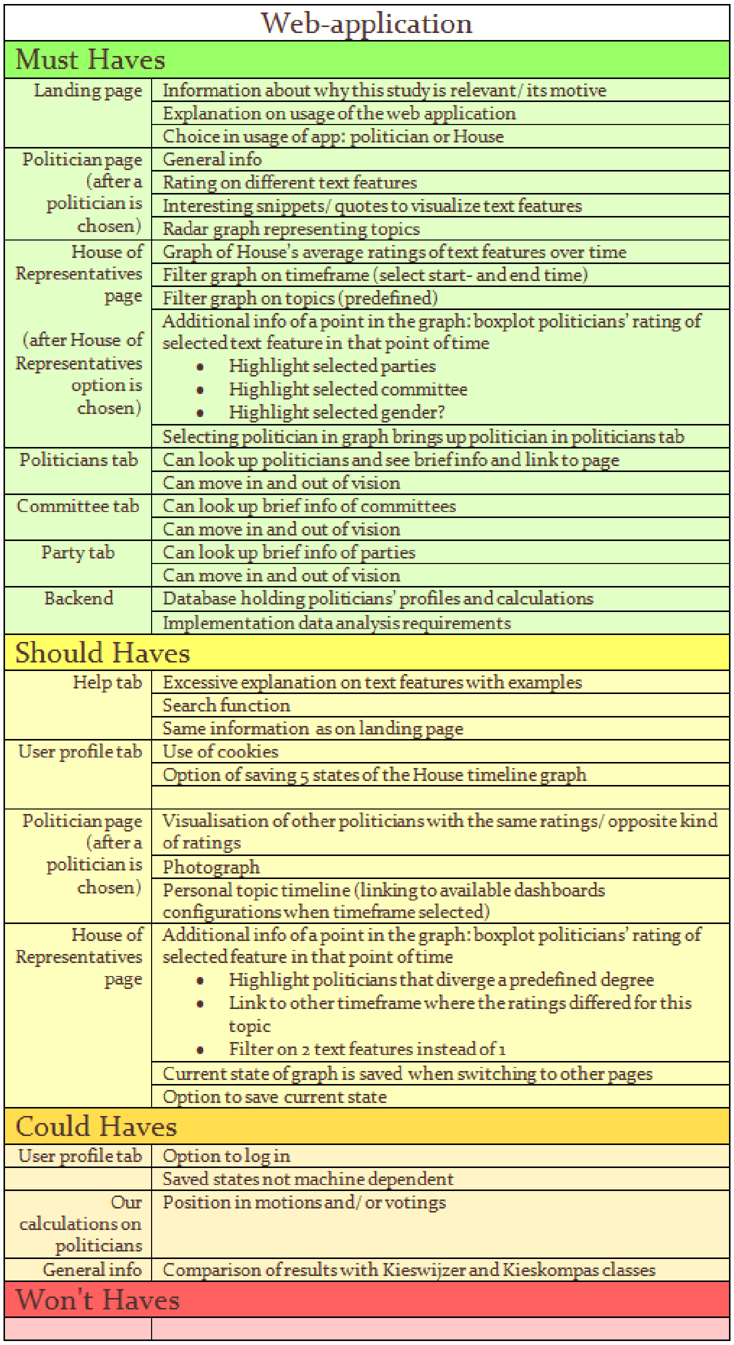
Figure 7 shows the planning as made during the first week of the project. Here, we have defined four different phases over the span of the project, and indicated the corresponding tasks. For each phase and task, we show the amount of weeks we were expecting to be necessary in order for us to finish them. These indicators proceed each phase and task, being respectively green and red. For each phase, the range of its indicator is thus defined by the start of the task we took on first, and the ending of the task we took on last. Below the phases, we mark the important deadlines during this project, as well as planned (group) meetings.

Figure 5: MoSCoW prioritization table for Data analysis



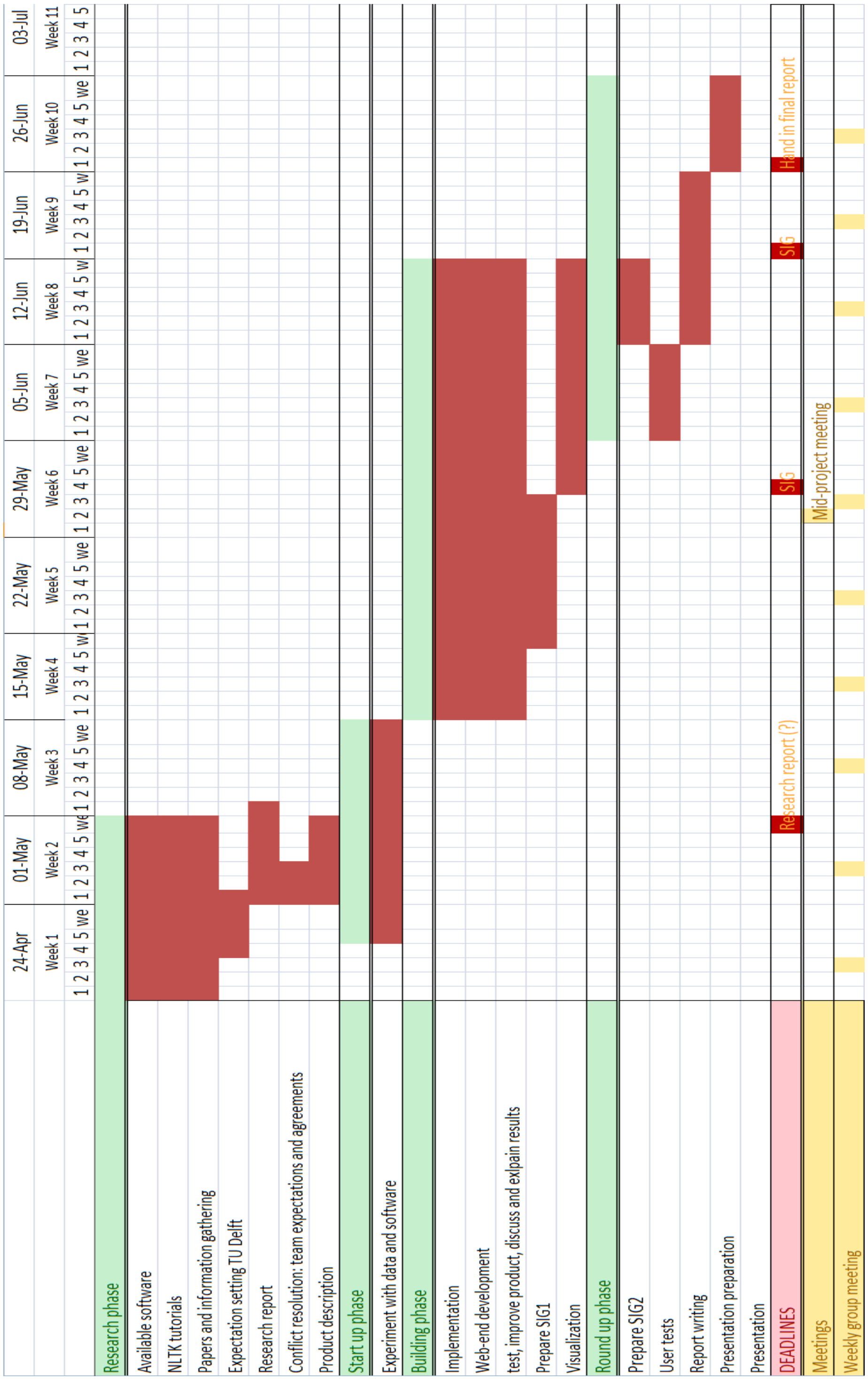
| | Data analysis | |
|---|---|---|
| | Must Haves | |
| | Data | Debate data retrieved via House of Representatives' official API |
| | | Cutting retrieved debate data according to speaker and giving it the following labels (keys in database):<br>• Speaker<br>• Date<br>• Activity (debate, questions round etc)<br>• Plenary or committee<br>Order number (to reconstruct original document order) |
| | | Processed data stored in (Amazon) database in following tables:<br>• Politician<br>• Party<br>• Committee<br>• Speech data<br>• Topics<br>• Events according to the political agenda<br>• Important (unforeseen) events |
| *Metadata* | General info politician | Basic profile: Name, committee, fraction/ party |
| | | Ratings on different text features |
| | | Amount of needed for radar graph |
| | | Interesting snippets for visualization |
| | | Ratings of text features |
| *Text features groups* | Sentiment analysis politician | Positive/ negative classifier |
| | Complexity analysis politician | Flesch-Kincaid score |
| | | Average length of sentences |
| | | Use of signal words |
| | | |
| *Metadata* | Needs for topics (timeline) | List of topics discussed |
| | | Topic's occurrences |
| | | Topics mapped to timeframes |
| | | Topic's mapped to politician/political party |
| | | House average on features in regard to debates/ topics |
| | Should Haves | |
| *Text features groups* | Manipulation analysis politician | Doublespeak rating (with snippet for visualization) |
| | | Active/ passive analysis |
| | Sentiment analysis politician | Active/ passive analysis |
| | | Correlation with complexity |
| | | Use of certain synonyms |
| | Feminine/ masculine analysis | Average rating |
| | | Rating during different events/ discussing certain topics |
| | Could Haves | |
| *Meta data* | Feminine/ masculine analysis | Interruptions done |
| | | Has been interrupted |
| | Won't Haves | |
| | | |

Figure 6: MoSCoW prioritization table for final product



| Web-application | |
|---|---|
| Must Haves | |
| Landing page | Information about why this study is relevant/ its motive |
| | Explanation on usage of the web application |
| | Choice in usage of app: politician or House |
| Politician page (after a politician is chosen) | General info |
| | Rating on different text features |
| | Interesting snippets/ quotes to visualize text features |
| | Radar graph representing topics |
| House of Representatives page<br>(after House of Representatives option is chosen) | Graph of House's average ratings of text features over time |
| | Filter graph on timeframe (select start- and end time) |
| | Filter graph on topics (predefined) |
| | Additional info of a point in the graph: boxplot politicians' rating of selected text feature in that point of time<br>• Highlight selected parties<br>• Highlight selected committee<br>• Highlight selected gender? |
| | Selecting politician in graph brings up politician in politicians tab |
| Politicians tab | Can look up politicians and see brief info and link to page |
| | Can move in and out of vision |
| Committee tab | Can look up brief info of committees |
| | Can move in and out of vision |
| Party tab | Can look up brief info of parties |
| | Can move in and out of vision |
| Backend | Database holding politicians' profiles and calculations |
| | Implementation data analysis requirements |
| Should Haves | |
| Help tab | Excessive explanation on text features with examples |
| | Search function |
| | Same information as on landing page |
| User profile tab | Use of cookies |
| | Option of saving 5 states of the House timeline graph |
| | |
| Politician page (after a politician is chosen) | Visualisation of other politicians with the same ratings/ opposite kind of ratings |
| | Photograph |
| | Personal topic timeline (linking to available dashboards configurations when timeframe selected) |
| House of Representatives page | Additional info of a point in the graph: boxplot politicians' rating of selected feature in that point of time<br>• Highlight politicians that diverge a predefined degree<br>• Link to other timeframe where the ratings differed for this topic<br>• Filter on 2 text features instead of 1 |
| | Current state of graph is saved when switching to other pages |
| | Option to save current state |
| Could Haves | |
| User profile tab | Option to log in |
| | Saved states not machine dependent |
| Our calculations on politicians | Position in motions and/ or votings |
| General info | Comparison of results with Kieswijzer and Kieskompas classes |
| Won't Haves | |
| | |

Figure 7: Initial planning



# 7 Development of the Data Analysis Tool

In this section we explain our approach on, and the structure of our final system. Our decisions regarding data retrieval, processing and visualisation can be found here. This chapter also marks the beginning of documenting the implementation phase.

For the backend of our application, we decided to use python. As mentioned in our research report, python offers a great variety of packages and modules regarding text analysis, and various of those full-filled some of our specific requirements. This was the main reason we chose this particular programming language. Eventually, it became clear that we did not depend as much on these packages as we initially thought, yet, python deemed itself to be of great use as we discovered tips and tricks that allowed us to quite efficiently reach our goal for the backend. Thus, data retrieval is also done with Python scripts.

## 7.1 Retrieval of Data

The House of Representatives is developing an API for easy access to their public data [10]. By registering we have received access to use this API and will be using its OData endpoint[11]. Documented House meetings that are retrievable online date from 1995 and onwards. To limit our data regarding topics and different politicians, we will start our analysis with a recent House with completed data: 2012 - 2017. Information is retrieved by means of (OData) queries and are returned in XML-format. The (shorthand) reports are docx. files. All data will be stored in a MariaDB database (see figure 9).

### Reading an processing the XML files from the API

The first step was to extract information about each member and the needed documents from the XML files. We obtained fields regarding personal information, referring the party he or she belonged to, the period of time the politician spend as a member etc. We also retrieved for each year a collection of 'raw' .docx files, which contained written debates, polls etc. from the House of Representatives. These were accompanied by several XML files that contained additional information about them, such as topic, or the date on which they were held. We also received XML files that held information about politician's that were or currently are members of the House of Representatives.

### Storing XML information

The parsed information from the XML files was then stored in a database running on a local MariaDB server. As we all have the most experience with SQL queries and this is open-source software, the choice is evident. The PyMySQL package was used as we decided on Python for the backend.

[10] https://opendata.tweedekamer.nl/

[11] http://www.odata.org/

#### Parsing the document files

The next step was parsing the .docx files as retrieved via the API, by transforming them into .txt files. For this, we used python's build in docx-module. The module allowed us to convert every .docx file into the desired .txt format. During the process, we had to slightly tweak each file, as the parser inserted or removed tokens that maintained structure in the file's .docx state.

#### Extracting politicians and assign their texts

Now that the text files were in the required format, we analysed their content and in particular how speakers were indicated. For each politician that we obtained via the XML files, we reconstructed their in-text indicators; preamble, last name, and the corresponding party between parentheses. For ministers, these were in the form of; Minister, last name. Using these indicators, we were able to determine which piece of text belonged to which politician. The difficult part here was that we split the text on each politician's name. Using python's build in split method, the string you split on gets lost as the text gets split into separate chunks. Because of this, we append the name of the politician to the new chunk. However, if some politician's name exists in such a chunk, this name possibly get added in front of another name; the result from a previous split. To solve this, we removed all but the politician's name with the highest index during the final step of the algorithm. The text should be assigned to this particular name, as it was the string the piece of text originally got split on.

## 7.2 Analysis of Data

Our features are divided into three categories, namely difficulty (or complexity), sentiment and masculine / feminine ratio. For the first category, we have implemented the following features; Flesch-Kincaid grade level, the average sentence length, and the percentage of signal words used. For the second category, we have implemented; positive / negative ratio, and the use of synonyms which results into an overall more positive or negative speech pattern of a politician. The last category refers to the feature itself.

### 7.2.1 Toolkits and resources

In this section, we name the toolkits and their including methods used for the implementation of our features. We also describe other necessary resources, such as additional dictionaries.

### NLTK

NLTK[12], or Natural Language Toolkit, is a platform that comes with many build-in packages and modules regarding language processing. These also include tools that allow sentiment analysis.

### LIWC dictionary

The LIWC dictionary contains 66 categories, and assigns these to each word that is included in its list. The categories range from grammar type to emotional value, if a word could refer to a hobby or work, etc. There are also just over 11.000 words, were every single one of them is proceeded by a (couple of) number(s) indicating the categories it is associated with.

### OpenTaal synonym list

The OpenTaal[13] synonym list contains nearly 300.000 lines, were each word is proceeded by a list of its synonyms.

## 7.2.2 Features

In this section, we give a (brief) description of how we implemented these features, and why we made certain decisions.

### Average sentence length (Difficulty)

This feature needed the total amount of words, and the total number of sentences in the text. We defined tokens that marked the end of a sentence; period, question mark and exclamation mark. So this results into:

$$\textit{Average sentence length} = \frac{\textit{Number of words}}{\textit{Number of sentences}}$$

### Signal words percentage (Difficulty)

For this feature we required a list of Dutch signal words, and computed the fraction of how often any of these signal words appeared in the text divided by the total amount of words. In terms of a formula:

$$\textit{Signal words percentage} = \frac{\textit{Number of signal words}}{\textit{Number of words}} \times 100\%$$

[12] http://www.nltk.org/

[13] https://www.opentaal.org/

### Flesch-Kincaid (Difficulty)

The Flesch-Kincaid grade level can be computed as:

$$\textit{FK grade level} = 0.39 \times \frac{\textit{Number of words}}{\textit{Number of sentences}} + 11.8 \times \frac{\textit{Number of syllables}}{\textit{Number of words}} - 11.59$$

(Note that this formula is used for text written in English, so as we use Dutch, the score might not be a perfect indicator, yet should make for a reasonable estimation as differences between different politicians can still be clearly visualized.) The total amount of words can be simply computed, as is the number of sentences. Counting the number of syllables in the text however, let alone in a single word, proved itself to be a more difficult matter. Each language has its own rules for when syllables can be formed within a word. We only came across a few options that would allow this calculation automatically, generally relying on NLTK, but these were all build around the English language. We then decided to use an approach similar to one we came across on the web[14]. Here it uses the Naive-Bayes classifier from NLTK, trained on features of a word, such its length, number of consonants etc. We ended with a classifier that made two wrong classifications, on an excerpt of a debate consisting of 166 words. In the discussion chapter, we talk a bit more about the eventual impact. Using the scikit-learn algorithm cheat-sheet[15], we also tried various other types of classifiers. We focussed mostly on the ones in the classification-field, as this was the type of goal we wanted to achieve. Many of these classifiers however seemed to be suitable only when there were just two possible categories, like determining if a text is positive or negative. As we wished to determine the amount of syllables in a word - a value that can range from one to possibly ten, - these type of classifiers did not work. The only classifier from this field that came remotely close to the predications the Naive-Bayes classifier made, was the K-Neighbours classifier - despite the indication that it should better be avoid for text data. Unfortunately, the latter made slightly more, but especially greater deviations than the Naive-Bayes classifier.

### Positive / Negative Ratio (Sentiment)

For this feature, we calculate a ratio between the total amount of 'positive' words, divided by the total amount of 'negative' words. So, in order to perform this calculation, we required a type of dictionary that shows if words are associated with respectively positive or negative emotions. The solution was the LIWC dictionary. We used this dictionary to count the number of words that were either in the 'posemo' or 'negemo' category, and returned the fraction between their total amount. As a formula:

$$\textit{Positive / negatve ratio} = \frac{\textit{Number of 'positive' words}}{\textit{Number of 'negative' words}}$$

[14] http://www.sociology-hacks.org/?p=128

[15] http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

### Synonym usage (Sentiment)

For this feature, we again used the LIWC for determining if words were associated with either positive or negative emotion. However, we also needed a list words with their corresponding synonyms. This came in the form of the OpenTaal synonym list. For each word in the input-text, we retrieved its synonyms and checked if the word itself and its synonyms were associated with emotional value, using the LIWC dictionary. We sought for the difference that could arise; if the word itself was associated with negative value, but it has a synonym with no emotional value, then the 'negative counter' increased by one, as the politician could've made a different choice of words in that case. An equal approach was taken in order to determine positivity. As a threshold, a valid synonym must have at least two categories matching the original word, so that it would be more likely that the synonym could be a replacement in the same context. The ratio can be expressed as:

$$\textit{Synonym Pos / Neg ratio} = \frac{\textit{Number of 'positive' synonyms}}{\textit{Number of 'negative' synonyms}}$$

### Feminine to Masculine speech style ratio

We calculated this ratio according to the method described in [8]. The author of this paper presented the feminine and masculine categories to be sets of other categories. In order to perform a calculation herself, she used the LIWC dictionary as well, and returned the fraction between the sum of all categories in the feminine set, and those within the masculine set. The method we used here was in fact the same as described in the 'Positive / Negative Ratio' paragraph, yet information about additional categories as well. In terms of a formula:

$$\textit{Feminine to masculine ratio} = \frac{\textit{Number of 'feminine' words}}{\textit{Number of 'masculine' words}}$$

## 7.3 Visualisation

In this section, we briefly discuss the tools we used in order to visualize the feature scores.

### 7.3.1 Web application design: Bootstrap

Bootstrap is an HTML, CSS and JavaScript framework for responsive and mobile first web development[16]. Bootstrap template providers offer beautiful interactive templates to design a web application. The template used for our product is called 'Superhero'[17] and available at the bootswatch website.

### 7.3.2 Data visualisation: D3.js

D3.js is a JavaScript library that allows you to visualise your data by binding it to a Document Object Model (DOM) and applying data-driven transformations[18]. Both the box-plot [19] and the time-line [20] charts are adapted from D3.js code to fit our data and web application and added.

### 7.3.3 Data retrieval: PHP

We used PHP in order to communicate with the database.

[16] http://getbootstrap.com/

[17] https://bootswatch.com/superhero/

[18] https://d3js.org/

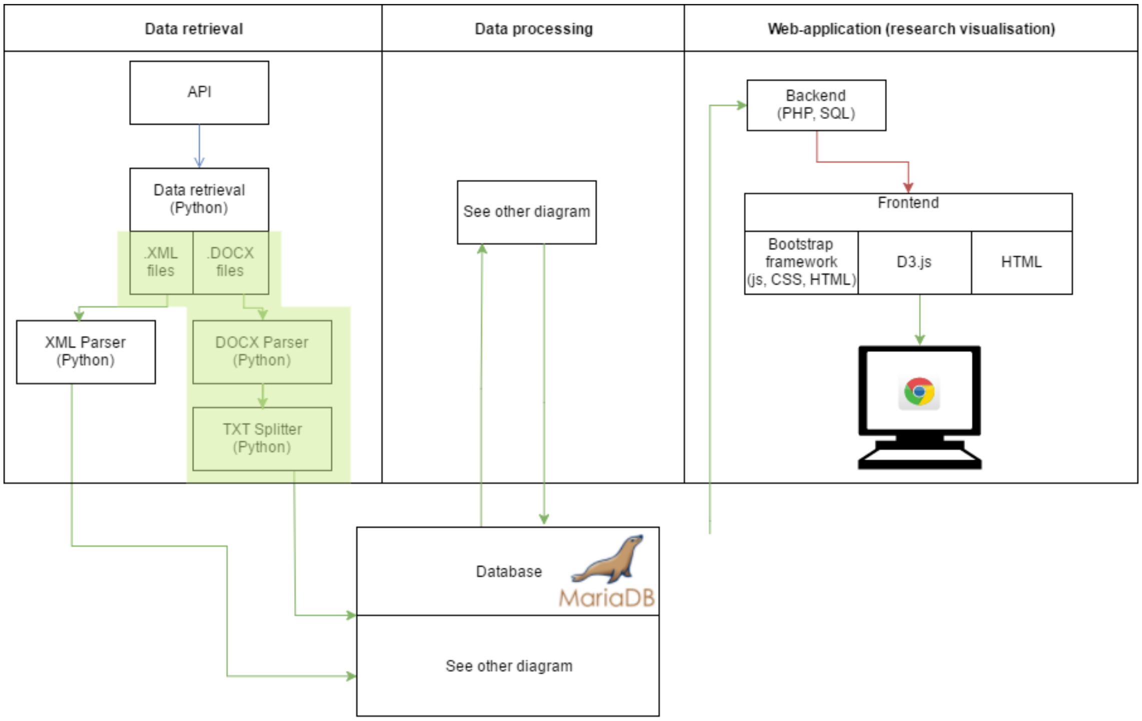
[19] http://mcaule.github.io/d3_exploding_boxplot/

[20] http://bl.ocks.org/DStruths/9c042e3a6b66048b5bd4

## 7.4 System Architecture

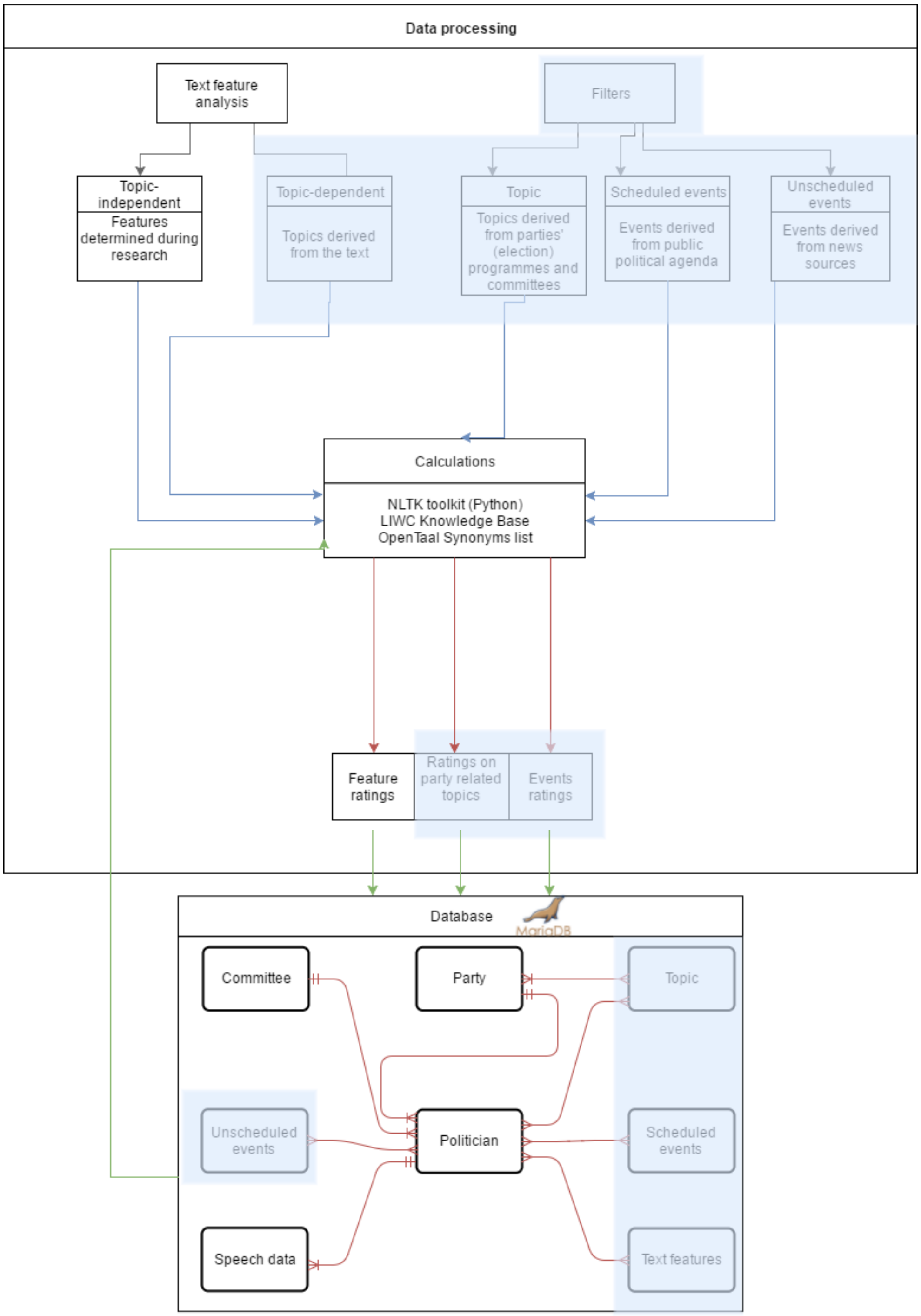
In figures 8 and 9 you will find an overview of our system's architecture. For visualising the relationships in the database, we use the Crow Foot Notation [21]. The elements highlighted in yellow were not considered at the start of implementation and were gradually added. The elements highlighted in purple were not added in the end, but present in the original idea.

Figure 8: Complete system's Architecture



[21] https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

Figure 9: System's Data flow



## 7.5 Testing

During the development process, we made use of several testing methods to improve the different components of our application. In this section, we will discuss which of these methods were applied to which component, and how these contributed to the overall product.

### 7.5.1 Code testing

For the features of our application, there was no real ground-truth for us to verify our results. It often came down to manually determine if our results were valid. In the case of our features, we wrote short texts that contained specific words for each feature, so that we could verify if they had the correct influence on the eventual ratio or percentage. For some of these results, we first determined which values we expected to see. For example, we were expecting the Flesch-Kincaid grade scores to be somewhere between seven and fourteen (a better estimated range as compared to the one mentioned in the research report). If any outliers were detected, we analysed the text causing these diversions, and then improved our code. This particular method lead to an upgrade of our Naive-Bayes classifier, which initially made more amount-of-syllable-per-word estimation errors compared to its current version. As an improvement here, we defined more properties to be extracted for each word on which the classifier was then trained upon. As it initially made twelve estimation errors for our test-text - which was an excerpt of a debate containing 166 words - it only made two after the addition. In general for each of the features, it was often counting the number of specific words, determining their corresponding categories if necessary, and then manually verifying the ratio or percentage.

#### Verifying score consistency

Once we got data for several years and could thus obtain multiple of the same type of score for each politician, we also verified the correctness of our results on these 'time-lines'. We were expecting values to just slightly increase or decrease over time, and in any other case, we again traced the origin of the cause. As mentioned in paragraph 9.4, we noticed that in 2017 there were some great deviations for several politicians regarding previous years. This lead to the discovery that this was not due to our features, but to a lack of input data for these particular politicians.

#### Performance testing

The aspect of performance testing came automatically as we received data from the API. We required to import all the text files per year at once, and then perform calculations in order to get values for our features. It was obvious that the algorithm did not get overloaded by an input of this size as each of the feature calculations were executed correctly. It could however be that perhaps some information may be lost during the parsing / reading or analysis itself, but with an amount this big, this might be impossible to tell.

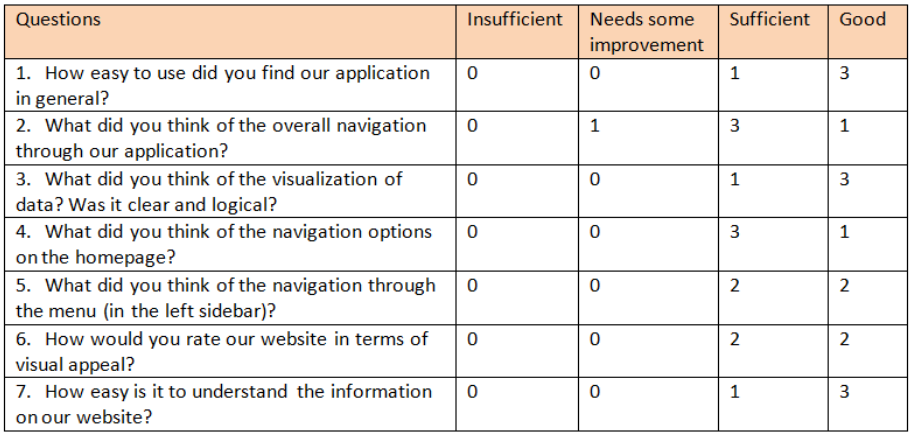
### Regression testing

For regression testing, we always compared the output of changed code or updated code to the output we deemed correct from a previous iteration. If it was the same, then no further changes were made. If this wasn't the case however, we again focussed on our smaller test-texts and determined whether the new code resulted into more erroneous (intermediate) behaviour, or actually fixed problems that went previously unnoticed. Every time a new feature was added that also used methods from other features or classes, we again verified the behaviour of all involved.

### 7.5.2 User testing

Another very important testing method was that of user testing. Our results and findings can be accessed via a web-application, so we want this to be clear and structured. In order for users to provide feedback, comments and remarks, we created a survey. In the table 10 below, we show the results of the four filled-in surveys we obtained. The numbers in each field indicate how often a particular score was given regarding the corresponding question. Since there was a total of four surveys, the frequencies in each row add up to four. As for the open questions; the named improvements were regarding the clarity of particular aspects on the website, and the navigation through multiple pages. The complete surveys can be found in appendix F.

Figure 10: usertests



| Questions | Insufficient | Needs some improvement | Sufficient | Good |
|---|---|---|---|---|
| 1. How easy to use did you find our application in general? | 0 | 0 | 1 | 3 |
| 2. What did you think of the overall navigation through our application? | 0 | 1 | 3 | 1 |
| 3. What did you think of the visualization of data? Was it clear and logical? | 0 | 0 | 1 | 3 |
| 4. What did you think of the navigation options on the homepage? | 0 | 0 | 3 | 1 |
| 5. What did you think of the navigation through the menu (in the left sidebar)? | 0 | 0 | 2 | 2 |
| 6. How would you rate our website in terms of visual appeal? | 0 | 0 | 2 | 2 |
| 7. How easy is it to understand the information on our website? | 0 | 0 | 1 | 3 |

# 8 Results

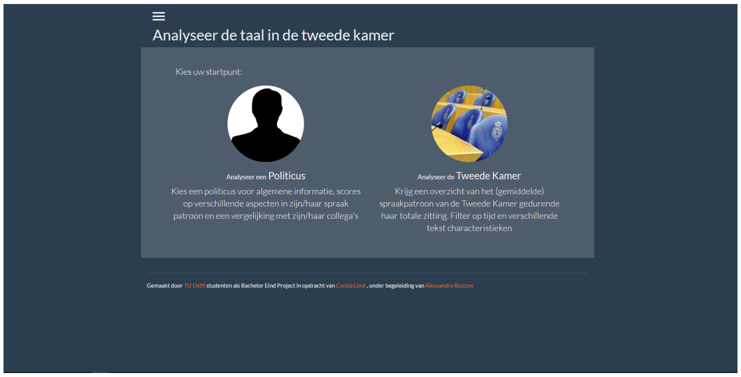
In this Results section we will first show screen-shots of the Data Analysis Tool (DAT) we developed with an explanation of the navigation and analysis options of the tool in section 8.1. In the second part, Section 8.2, we included box-plots and scatter plots of our data to evaluate the mean values of all features.

## 8.1 Data Analysis Tool

The DAT is designed in such a way it guides the user through the data. In figure **??** the landing page is shown; where the user starts. He/ she is given the choice of exploring the data of an individual Representative or that of the whole house.

### 8.1.1 Analysing a Representative

By choosing the former, the user can search for a Representative from a list : figures 19 and 13. After selection the user arrives at the respective page.



Analyseer de taal in de tweede kamer

Kies uw startpunt:

Analyseer een Politicus

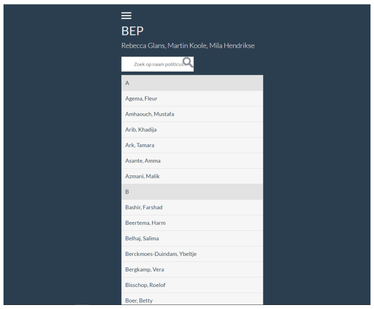
Kies een politicus voor algemene informatie, scores op verschillende aspecten in zijn/haar spraak patroon en een vergelijking met zijn/haar collega's

Analyseer de Tweede Kamer

Krijg een overzicht van het (gemiddelde) spraakpatroon van de Tweede Kamer gedurende haar totale zitting. Filter op tijd en verschillende tekst characteristieken

Gemaakt door TU Delft studenten als Bachelor Eind Project in opdracht van Cyntia Liem , onder begeleiding van Alessandro Bozzon

Figure 11: The landing page



BEP

Rebecca Glans, Martin Koole, Mila Hendrikse

Zoek op naam politicus..

A

Agema, Fleur

Amhaouch, Mustafa

Arib, Khadija

Ark, Tamara

Asante, Amma

Azmani, Malik

B

Bashir, Farshad

Beertema, Harm

Belhaj, Salima

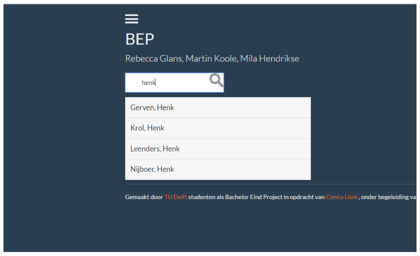
Berckmoes-Duindam, Ybeltje

Bergkamp, Vera

Bisschop, Roelof

Boer, Betty

Figure 12: Arriving at the selection page



BEP

Rebecca Glans, Martin Koole, Mila Hendrikse

henk

Gerven, Henk

Krol, Henk

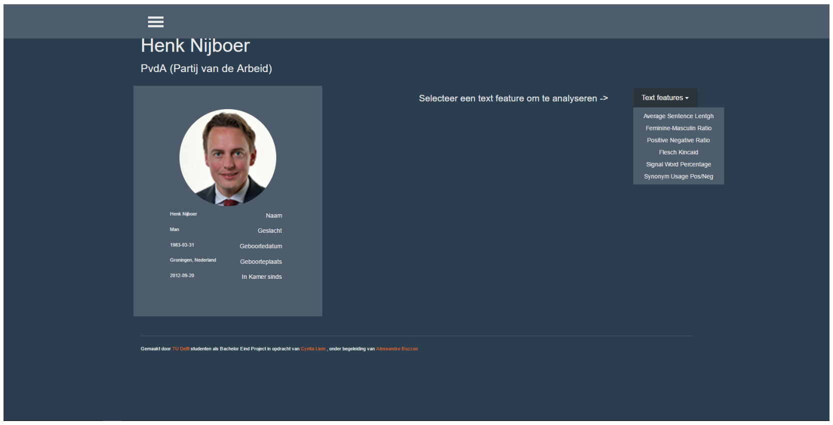
Leenders, Henk

Nijboer, Henk

Gemaakt door TU Delft studenten als Bachelor Eind Project in opdracht van Cyntia Liem , onder begeleiding va

Figure 13: Selecting a Representative

The Representative's page (politicus pagina) holds general information and a personal time-line. The timeliness is used to view the Representative's text feature ratings over time, which can be seen in figures **??** and **??** . By selecting a text feature, the Representative's ratings appear.



Henk Nijboer

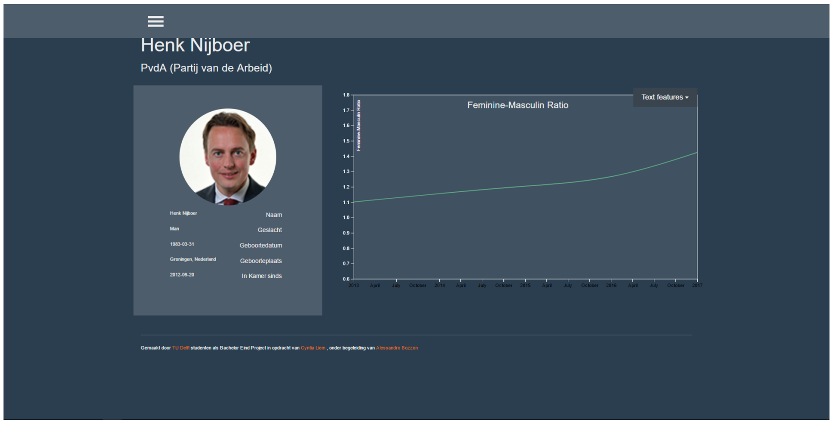
PvdA (Partij van de Arbeid)

| | |
|---|---|
| Henk Nijboer | Naam |
| Man | Geslacht |
| 1983-03-31 | Geboortedatum |
| Groningen, Nederland | Geboorteplaats |
| 2012-09-20 | In Kamer sinds |

Selecteer een text feature om te analyseren ->

Gemaakt door TU Delft studenten als Bachelor Eind Project in opdracht van Cyntia Liem , onder begeleiding van Alessandro Bozzon

Figure 14: The landing page



Henk Nijboer

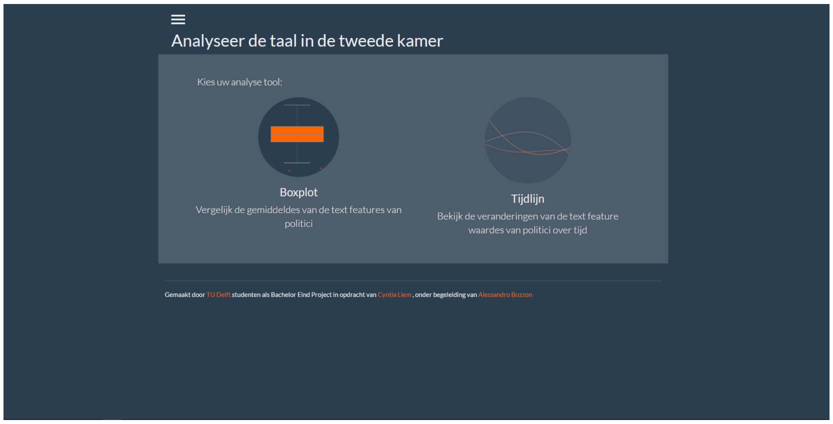
PvdA (Partij van de Arbeid)

| | |
|---|---|
| Henk Nijboer | Naam |
| Man | Geslacht |
| 1983-03-31 | Geboortedatum |
| Groningen, Nederland | Geboorteplaats |
| 2012-09-20 | In Kamer sinds |

Gemaakt door TU Delft studenten als Bachelor Eind Project in opdracht van Cyntia Liem , onder begeleiding van Alessandro Bozzon

Figure 15: The landing page

If our user had chosen to analyse the complete DHoR, he/ she would have arrived at a second choice page. Would he/ she want to see the data represented in either box-plots or time-lines?



Analyseer de taal in de tweede kamer

Kies uw analyse tool:

Boxplot

Vergelijk de gemiddeldes van de text features van politici

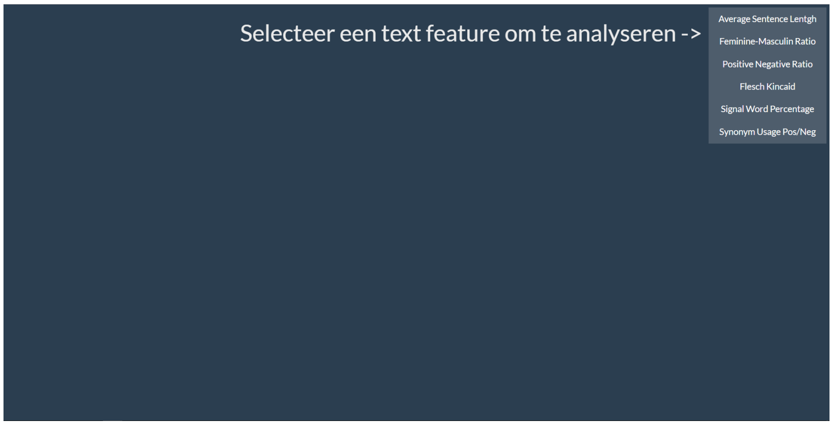
Tijdlijn

Bekijk de veranderingen van de text feature waardes van politici over tijd

Gemaakt door TU Delft studenten als Bachelor Eind Project in opdracht van Cyntia Liem , onder begeleiding van Alessandro Bozzon

Figure 16: Option page for analysing the complete DHoR

Choosing the time-line option results in a page with a time-line similar to the one in the Representative's page. Once a text feature is selected, the user can select multiple Representatives on the right to plot simultaneously.



Selecteer een text feature om te analyseren ->

Figure 17: Analysing the complete DHoR by time-line

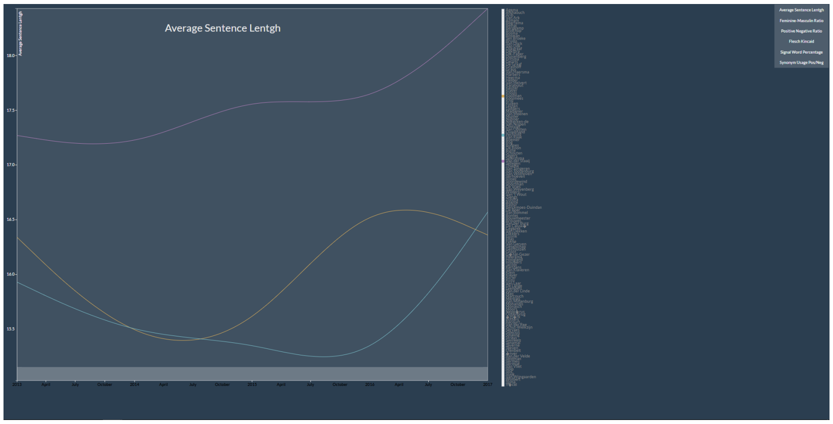


Figure 18: Choosing several politicians

Once the user has filled the time-line, he can select a time-frame at the bottom of the plot. This time-frame can be used to zoom in and explore the data over time.

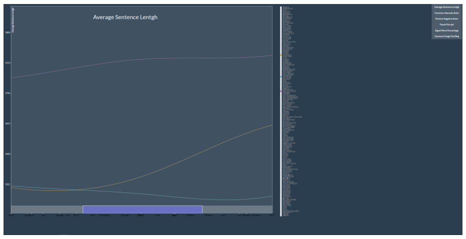


Figure 19: Selecting a time-frame

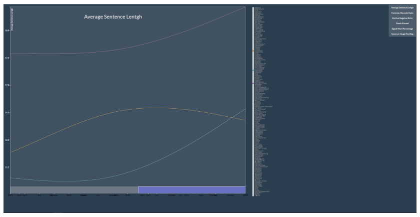


Figure 20: Scrolling with the selected time-frame

Choosing the box-plots results in a page holding a box-plot for each text feature. The data in the box-plots represent each Representative's average rating. Clicking on a box-plot reveals the underlying data points, hovering on these data-points gives the name of the corresponding Representative.

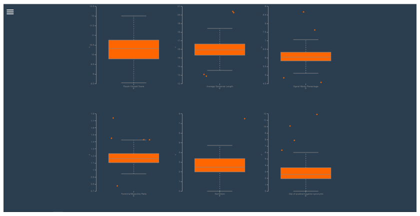


Figure 21: The boxplots page

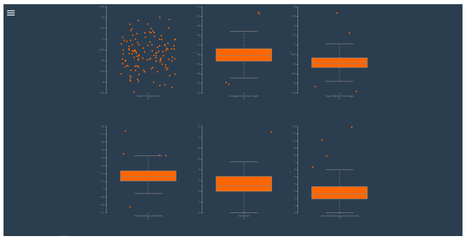


Figure 22: Revealing the underlying datapoints

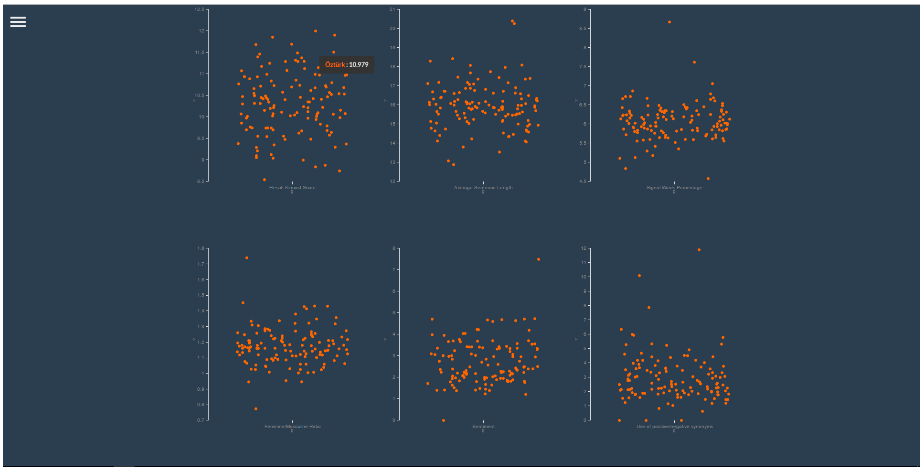
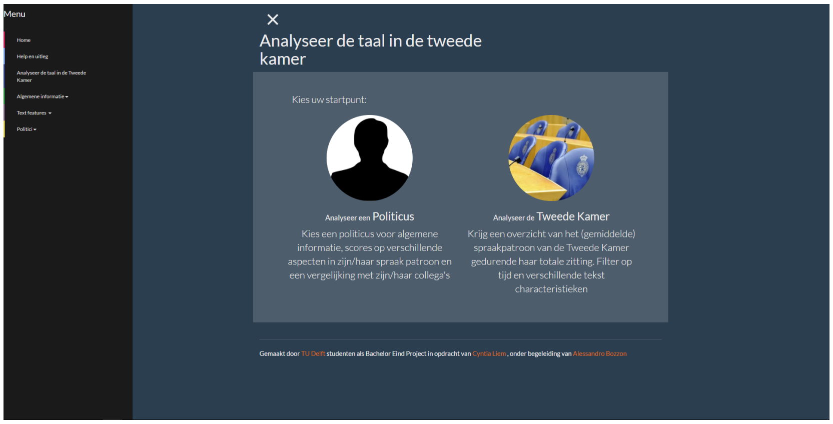


Figure 23: Getting the name of the Representative

Navigating between different pages can be done with the sidebar. This is revealed by clicking the “hamburger”-icon at the top left of the screen. In this sidebar the user can find different information regarding the tool, including an explanation on the text features. The list of Representatives is present as well, making navigating to other Representatives easier.



Analyseer de taal in de tweede kamer

Kies uw startpunt:

Analyseer een Politicus

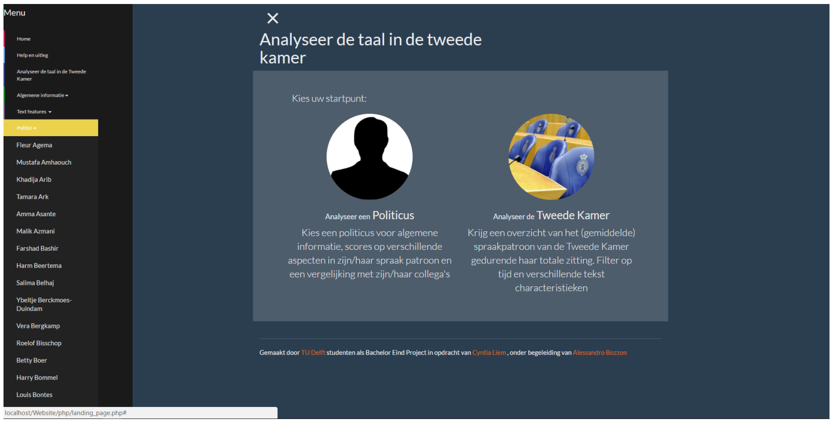
Kies een politicus voor algemene informatie, scores op verschillende aspecten in zijn/haar spraak patroon en een vergelijking met zijn/haar collega's

Analyseer de Tweede Kamer

Krijg een overzicht van het (gemiddelde) spraakpatroon van de Tweede Kamer gedurende haar totale zitting. Filter op tijd en verschillende tekst characteristieken

Gemaakt door TU Delft studenten als Bachelor Eind Project in opdracht van Cyntia Liem , onder begeleiding van Alessandro Bozzon

Figure 24: Revealing the sidebar



Analyseer de taal in de tweede kamer

Kies uw startpunt:

Analyseer een Politicus

Kies een politicus voor algemene informatie, scores op verschillende aspecten in zijn/haar spraak patroon en een vergelijking met zijn/haar collega's

Analyseer de Tweede Kamer

Krijg een overzicht van het (gemiddelde) spraakpatroon van de Tweede Kamer gedurende haar totale zitting. Filter op tijd en verschillende tekst characteristieken

Gemaakt door TU Delft studenten als Bachelor Eind Project in opdracht van Cyntia Liem , onder begeleiding van Alessandro Bozzon

Figure 25: Selecting a Representative in the sidebar

## 8.2 Data Results

In Table 1 the abbreviations of the six implemented text features, which are used in the charts to follow, are shown.

| Text-feature name | Abbreviation |
|---|---|
| Flesch-Kincaid | FK |
| Average Sentence Length | ASL |
| Signal Words Percentage | SWP |
| Feminine to masculine speech style ratio | FM |
| Positive/Negative Ratio | PNR |
| Synonym usage | Synon. |

Table 1: Abbreviations used in results table

In Figure 26 the boxplots are shown of the values per text feature of all Representatives. The values to plot the boxplots are shown in Table 2.

| | FK | ASL | SWP | FM | PNR | Synon. |
|---|---|---|---|---|---|---|
| Minimum | 8.542 | 12.881 | 4.575 | 0.774 | 0.000 | 0.000 |
| 25th percentile - Minimum | 1.238 | 2.431 | 1.249 | 0.331 | 1.989 | 1.928 |
| Median - 25th percentile | 0.537 | 0.612 | 0.219 | 0.057 | 0.486 | 0.665 |
| 75th percentile - Median | 0.436 | 0.689 | 0.286 | 0.072 | 0.891 | 1.044 |
| Maximum - 75th percentile | 1.246 | 3.787 | 2.344 | 0.506 | 4.127 | 8.267 |

Table 2: Minimum, 25th percentile, median, 75th percentile and maximum values of each text feature

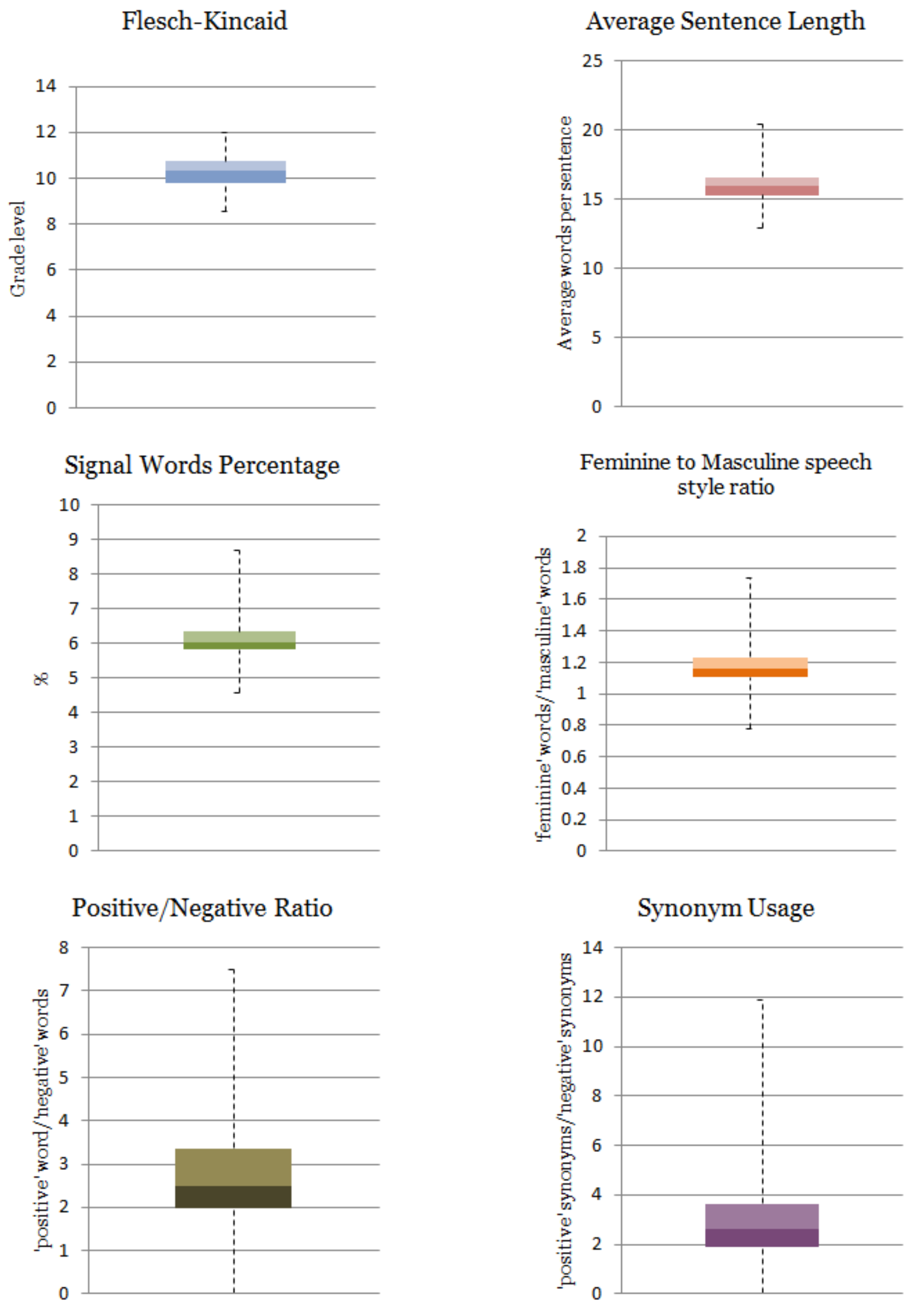


Figure 26: Boxplots of values for the six text features for all Representatives

In figure 27 the FM values of male and female Representatives are plotted in a boxplot separately.

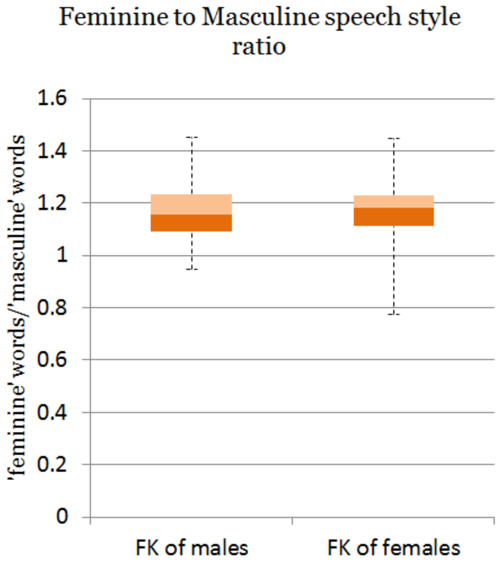
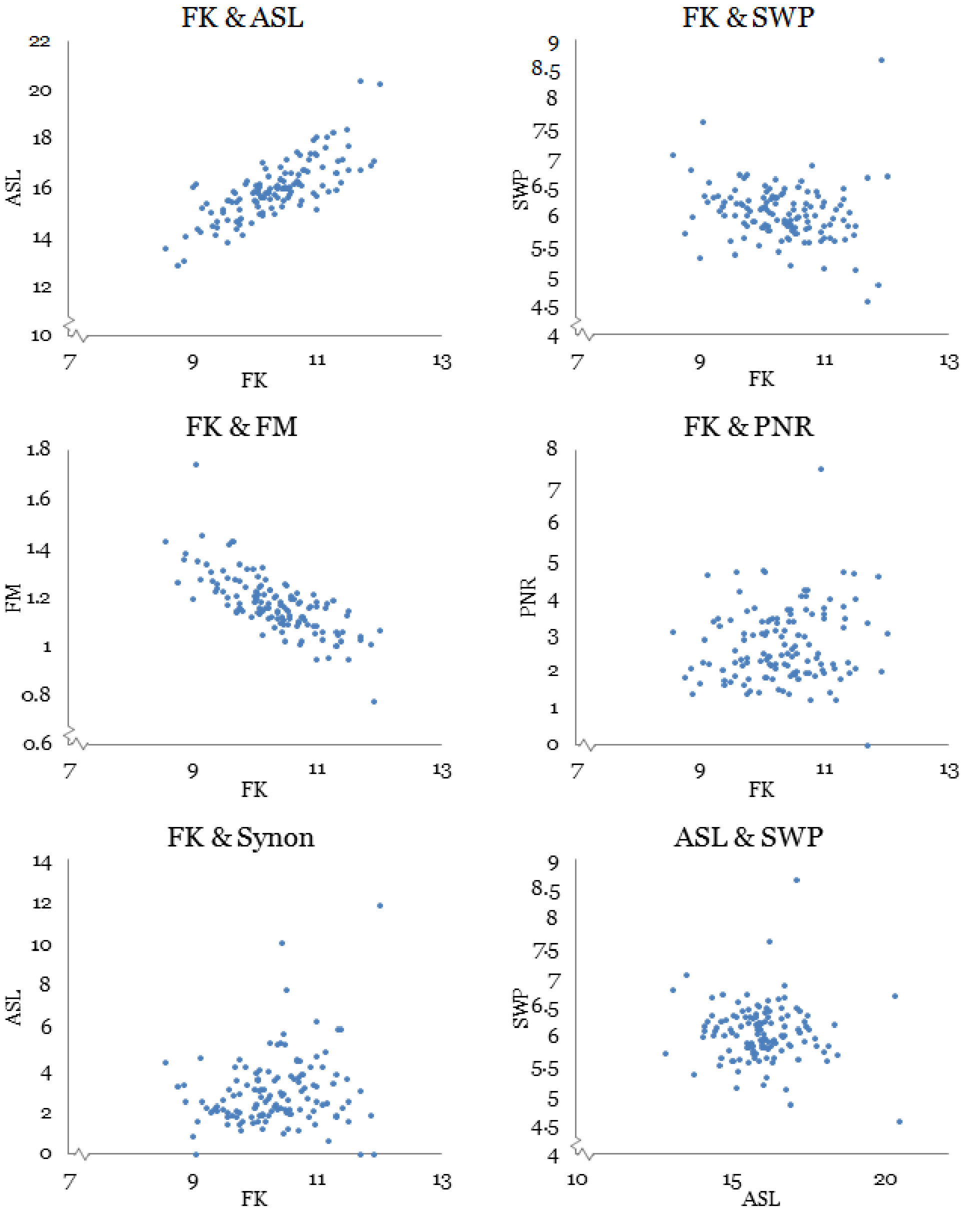


Figure 27: Boxplots of the FM text feature: comparison between the values of the female and male Representatives

According to the combination formula:

$$C(n,k) = \frac{n!}{(n-k)!k!} = \frac{6!}{(6-2)!2!} = 15$$

where $n = 6$ and $k = 2$, with six text features there are 15 possible pair combinations.In figures 28, 29 and 30 you find all scatter-plots of the combination of the results of each text feature pair.



FK & ASL

FK & SWP

FK & FM

FK & PNR

FK & Synon

ASL & SWP

Figure 28: Scatter plots of combination of text feature results

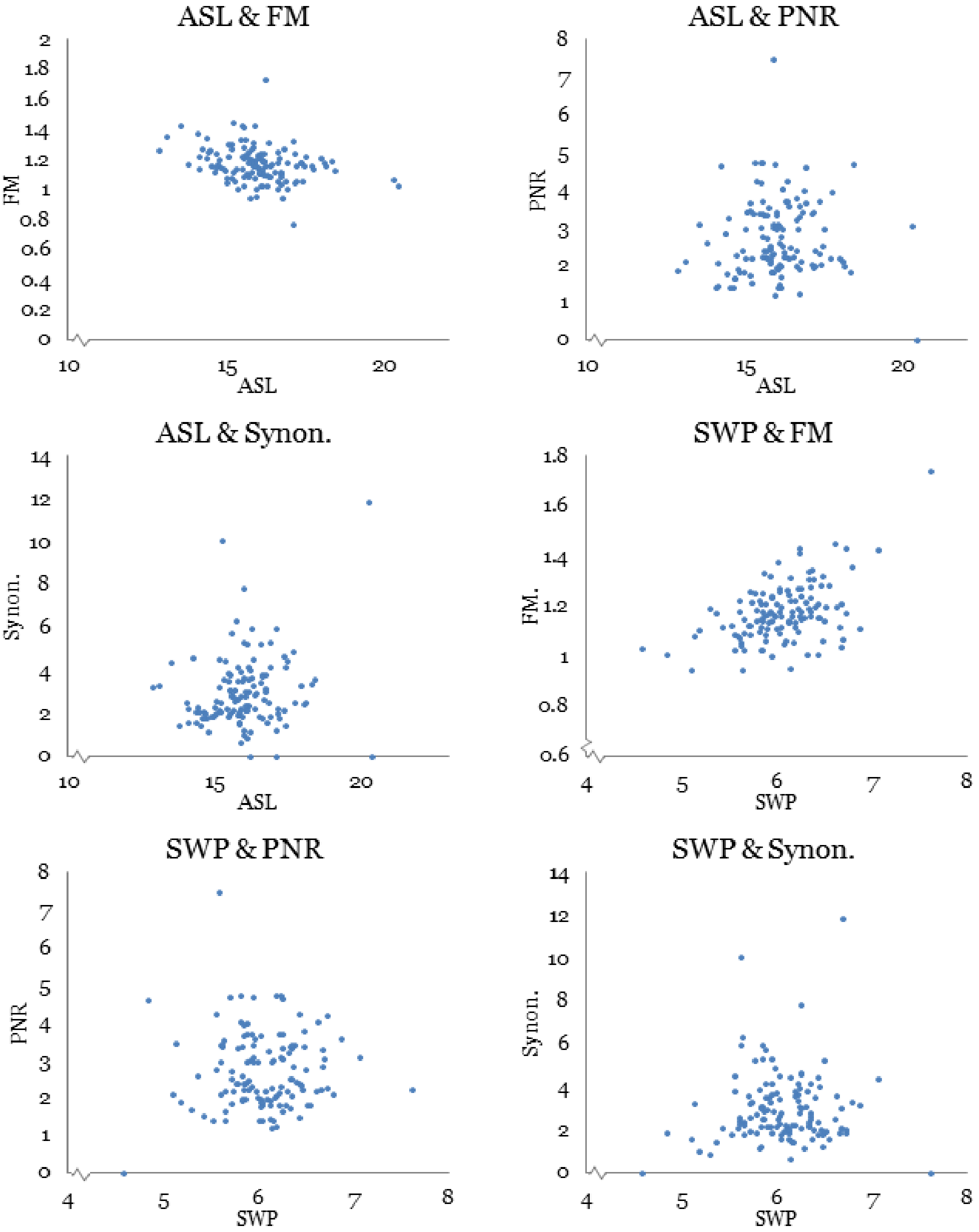


Figure 29: Scatter plots of combination of text feature results

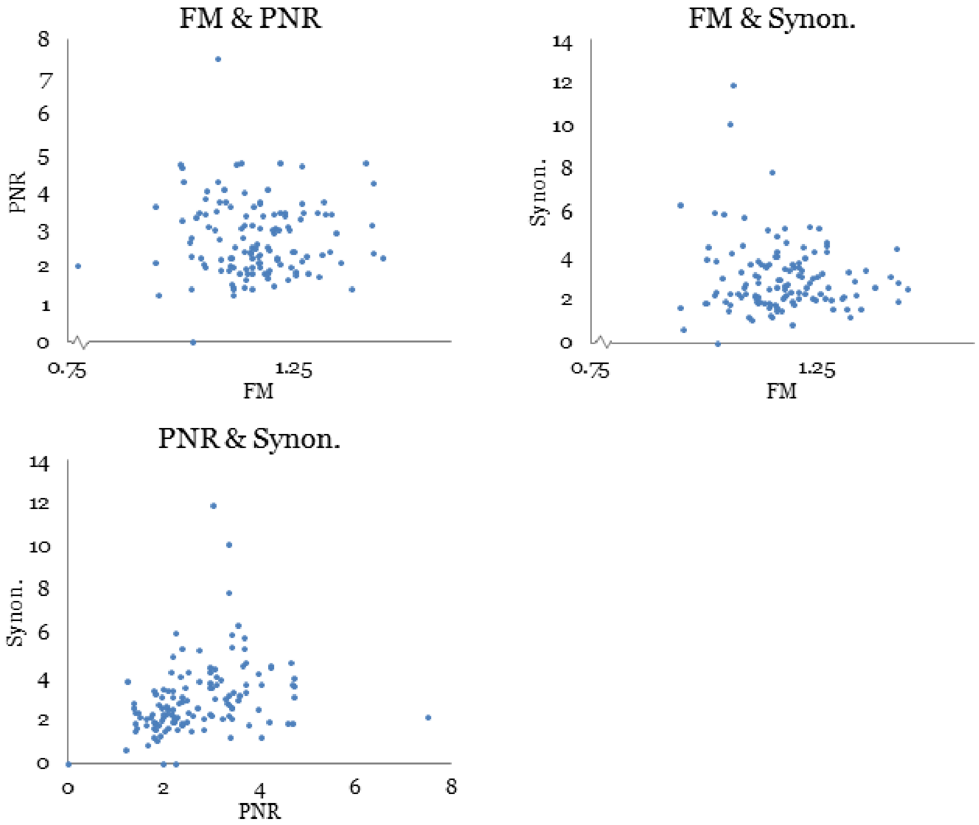


Figure 30: Scatter plots of combination of text feature results

To be able to say something about the correlation between features, we calculated the *correlation coefficient* [11]:

$$Correl(X,Y) = \frac{\sum(x-\overline{x})(y-\overline{y})}{\sqrt{\sum(x-\overline{x})^2\sum(y-\overline{y})^2}}$$

*where X is set 1 and Y is set 2*

of each pair of text feature result sets. The results of the calculations are shown in Table 3. The correlation coefficient (*Correl*) measures the strength and direction of a linear relationship between two variables on a scatterplot. It can take up a value between -1 and 1, where -1 is a perfect negative linear relationship (and values close to -1 suggest a strong negative relationship), 0 means no linear relationship between to datasets, and 1 is a perfect linear relationship (values close to 1 suggest a strong positive linear relationship).

| Set1 | Set2 | Correlation Coefficient |
|---|---|---|
| FK | ASL | 0.770278116 |
| FK | SWP | -0.181349254 |
| FK | FM | -0.738902472 |
| FK | PNR | 0.116102491 |
| FK | Synon. | 0.19356196 |
| ASL | SWP | -0.084091539 |
| ASL | FM | -0.302995979 |
| ASL | PNR | 0.017358469 |
| ASL | Synon. | 0.183734246 |
| SWP | FM | 0.281626366 |
| SWP | PNR | -0.007144313 |
| SWP | Synon. | -0.029276156 |
| FM | PNR | -0.011642239 |
| FM | Synon. | -0.104598314 |
| PNR | Synon. | 0.319492161 |

Table 3: Correlation coefficient between the sets of two text feature result of all Representatives

# 9 Discussion

In this chapter, we will discuss our results, and what these exactly mean.

## 9.1 Boxplots

When we look at the box-plots we see little variation between Representatives when looking at Flesch-Kincaid, Average Sentence Length, Signal Words Percentage and Feminine to Masculine speech style ratio. When the American school system is translated into the Dutch school system [22] a grade 10 level in the United States translates to a grade level 4 of high school in the Netherlands (translation: klas 4 van de middelbare school). The teenagers who attend the fourth grade of high school in the Netherlands are around 15 years old. So according to our calculations, the average complexity of language of the Representatives is similar to a 15 year old High School attendee's reading and writing skills.

The Average Sentence Length and Signal Words Percentage also don't vary much in the DHoR. Since we know Representatives have had to excel during their careers in order to enter the DHoR, we expect them to be able to communicate on the same intellectual level. Little variety in complexity measuring text features does therefore not surprise us.

The difference between the female and male average values of the Feminine to Masculine Speech Style ratio is negligible. There are a couple of possible explanations for this. The first one being that women in the Netherlands don't feel the pressure to 'talk like a man' when entering the (mostly) male environment of the DHoR as described in the paper *Talk "Like a Man": The Linguistic Styles of Hillary Clinton* by J. J. Jones[8]. The second possible explanation is that the Netherlands is an emancipated country and there is not as much culture division between men an women as in other countries.

## 9.2 Scatter plots

With the 15 scatter plots and the correlation coefficient table we aimed to find some correlation between all possible text feature pairs. The most significant correlation coefficients are visible in Table 4. It seems that having a high Flesch-Kincaid value correlates with a high Average Sentence Length value (*0.770278116*). There is a good explanation for this: the length of sentences plays a part in the calculation of the Flesch-Kincaid (7.2.2).

The second outstanding correlation (*-0.738902472*) we found was the one between Flesch-Kincaid and the Feminine to Masculine Speech Style ratio. If the latter is low, this indicates that the text verges toward masculinity. A category that belongs to the masculine category set is that of 'big words', i.e.words with a length greater than six. Since these 'big words' are more likely to contain more syllables as compared to shorter words, this might be the reason for the

[22] http://www.barthokriek.nl/ondertiteling/lijsten/schoolsysteem_VS.htm

correlation between the Flesch-Kincaid grade level score (as on average more syllables result into a higher score), and the Feminine to Masculine Speech Style ratio.

| Set1 | Set2 | Correlation Coefficient |
|---|---|---|
| FK | ASL | 0.770278116 |
| FK | FM | -0.738902472 |

Table 4: Correlation coefficient between the sets of two text feature result of all Representatives: most significant results

## 9.3 Expectations

In table 5 we show the average values for each text feature per Representative for the most important Respresentatives from each important party. We entered this Bachelor's thesis with some expectations about mostly one famous Dutch politicians and Representative: Geert Wilders, founder and leader of the party 'PVV' (short for Partij voor de Vrijheid, translation: Party for Freedom). He is a controversial figure with extreme opinions about immigration and the Islamic religion. We know him as a public figure who tends to make simplistic remarks about complex problems and is often (if not always) negative about any solution presented by the other parties. Therefore, we expected him to score low for the sentiment analysis text features values and indeed: he scores very low with a Positive/Negative ratio of *1.390* (*1.331* lower than the overall average). Also he is known for making simplistic and populist statement, which made us expect low values for Flesch-Kincaid and Average Sentence Length. Indeed, when comparing him to other important Representatives his values are the lowest for these two features. Regarding Geert Wilders, our Data Analysis Tool represented our expectations well.

Since the Flesch-Kincaid represents the knowledge of the Dutch language of the speaker in a way, we expected the Representatives who were born abroad to have a lower average value for Flesch-Kincaid. There were 13 Representatives in the DHoR in the past five years who are born abroad , and their average Flesch-Kincaid value is *10.252021761167617* which is only *0.0559061951* lower than the overall average. So country of birth doesn't appear to have any influence on the Flesch-Kincaid values.

| | FK | ASL | SWP | FM | PNR | Synon. |
|---|---|---|---|---|---|---|
| Geert Wilders (PVV) | 8.878 | 14.070 | 6.0115 | 1.380 | 1.390 | 2.550 |
| Fleur Agema (PVV) | 9.351 | 14.114 | 6.192 | 1.225 | 2.061 | 2.260 |
| Halbe Zijlstra (VVD) | 9.127 | 15.193 | 6.607 | 1.452 | 2.193 | 2.518 |
| Barbara Visser (VVD) | 10.351 | 16.128 | 6.490 | 1.142 | 2.755 | 5.238 |
| Henk Nijboer (PvdA) | 9.746 | 14.610 | 6.722 | 1.175 | 2.263 | 2.035 |
| Khadija Arib (PvdA) | 9.686 | 14.358 | 6.667 | 1.210 | 2.847 | 2.056 |
| Marianne Thieme (PvdD) | 11.396 | 17.179 | 6.087 | 1.0595 | 1.976 | 2.244 |
| Pieter Omtzigt (CDA) | 10.524 | 15.983 | 5.823 | 1.147 | 1.915 | 1.243 |
| Mona Keijzer (CDA) | 9.985 | 15.517 | 5.826 | 1.210 | 3.353 | 3.127 |
| Jesse Klaver (GroenLinks) | 9.745 | 15.588 | 5.860 | 1.334 | 2.385 | 2.866 |
| Linda Voortman (GroenLinks) | 10.567 | 16.652 | 5.992 | 1.209 | 2.967 | 3.719 |
| Emile Roemer (SP) | 9.676 | 15.802 | 6.213 | 1.273 | 2.144 | 4.189 |
| Renske Leijten (SP) | 9.972 | 15.840 | 6.139 | 1.318 | 2.279 | 1.519 |

Table 5: Average values for each text feature per Representative. In this table we included (where available) results of one important male and one important female member per party

# 10 Reflection on process

In this chapter, we discuss which obstacles we discovered during the implementation process, and how these may or are affecting our eventual results.

## 10.1 Data Retrieval

In the next section, we will discuss the problems and obstacles discovered regarding the data obtained via the API.

### The retrieved data from the API itself

The structure of the XML files retrieved via the API was very unclear. OData mentions every possible relation between entities making the files undesirable large and sequentially complicating the relations. This made them hard to analyse and in finding the information we needed. The information that (supposedly) matched was scattered across a politician's general entry field . (An entry fields contains all of a politician's information in many sub-fields, whereas one single entry was around 500 lines in length.) This made it difficult to sort out and group information efficiently. Unfortunately, the ministers' information were missing, which had to be added manually in order for our algorithm to function correctly. Even though the API is an automated way of retrieving data published on the House's website, as it is still under development we can not treat this data as official.

### Parsing the document files

As mentioned earlier, the data retrieved from the API covered each year from 2012 to 2016, and 2017 partially. However, after applying the parser to each file, it became clear that a lot of them were either unreadable or simply corrupted. Namely, all files between 2012 and 2013 were unusable. Later years also held a numerous amount of such files, yet this number became less with each proceeding year. We also stated that there had to be made some tweaks to each file. This was because when being parsed to a .txt file, some tokens indicating newlines ("\n") appeared in the text. On the other hand, spaces sometimes disappeared after periods, commas and other similar characters. For our search and analyse functions to perform sufficiently, these inconsistencies had to be dealt with.

### Extracting politicians and assigning their texts

When we tried to split text using a replicated indicator, it became clear that some of the politicians did not get any text assigned to them, contrary to our expectation. This means that a politician's values are not based on an equal amount of text. As it turned out, for some parties, the full party name was used, as opposed to the assumed use of abbreviation. Since we did not know for how many parties this was the case, we decided to also include indicators were the full party name was used. This solution may not seem perfect, but because of these kind of inconsistencies in the text file, we were not sure whether this problem could arise for a single politician as well. As mentioned, various politicians were missing from the XML files. We added these ones manually, but it is possible that the algorithm comes across names that it 'does not recognize', and therefore the text will not be split on that particular name. This means that the algorithm might erroneously link pieces of text to the wrong politician. A similar issue occurs because of the structure of the .txt files themselves. Sometimes, a politician_1 mentions another politician_2 in a way of; 'This is due to the mismanagement of Minister Rutte:', followed by an explanation by politician_1. However, the string 'Minister Rutte:' is used as an indicator in our algorithm, again introducing the possibility of wrong text assignation.

## 10.2 Features and resources

In the next section, we will discuss the problems and obstacles discovered regarding the resources used and implementation of the features.

### Flesch-Kincaid

The classifier in general thus did a great job on estimating how many syllables there were in each word in the text. However, as its accuracy wasn't a full 100%, the overall Flesch-Kincaid score might differ a little from its theoretical perfect value. Running the algorithm over a large amount of text – 100.000+ words–, resulted into slight deviations each time. These however were always somewhere between 0.0 and 0.1, often closer to 0.0.

### LIWC dictionary

In the LIWC dictionary, it is common that for verbs only their stem is shown, followed by a * token. The * indicates that multiple word 'endings' are possible, but since this might differ for each word, we decided not to use a brute-force-like approach by appending every possible combination to each word. Instead, we split the dictionary into two categories; words without a *, and words with a *. For each word in the text, we first checked if it was present in the first category. If this was the case, we could simply retrieve the corresponding categories. If this wasn't the case however, we repeatedly removed the last letter from the word a fixed amount of times and verified if it became any of the words in the second category (with the * removed of course). This allowed us to categorize almost all words in the text. Some words however were simply missing, which also was a recurring issue in the synonym feature.

## 10.3 General remarks on the encountered obstacles

Despite the discovery of a lot of unusable files, we still had plenty of information for each year - except year 2012-2013, and arguably 2017, as several politicians had very little speech (a mere couple of sentences) which resulted into great diversions from their previous (score-wise consistent) years. Regarding the previous, we therefore hope that the in some cases extremely large amounts of texts will approximately nullify the effect of incorrect assignation, or for example the erroneous predication of our classifier as used in the Flesch-Kincaid grade score calculation.

## 10.4 Ethics

Some ethical issues may arise due to the results we obtained. Here, we consider the following cases that might introduce ethical concerns;

- Users might generalize our results and jump to hasty conclusions.
- Users might use these values as an overall replacement.
- Users might forget the purpose of this application.
- Users might use scores to support their arguments.

The first case refers to what could happen if users see the difference in scores between several politicians. For example, if they come across politicians that have a below average Flesch-Kincaid grade score, they might label these members to be less intelligent, and therefore not suitable for taking place in the cabinet. In other words, people might tend to change their opinion about someone, without further questioning.

The second is in some way an extension to the first; people might assume that these mere values are enough to create a reasonable representation of a politician. These results are meant to give minimal insight in a person's speech pattern and can then be used to give the user insight in what appeals to him/ her. Again, a politician with favourable scores would probably be more appealing as compared to another with lower scores. As mentioned before, speech does not contain only textual aspects (of which we have not implemented all), but voice aspects as well. These results thus do not hold all values that could be of influence in a listener.

Again continuing on previous issues mentioned, the third case describes the possible lack of comprehension of our application. We implemented a speech analyser. The features visualized on our website are those we considered to be insightful when it comes to analysing the text in the speech of a politician. It does therefore not say anything about the topics addressed, which types of arguments were made, or if a politician made any sense at all. Users of our application should realize that these features do not directly address the content of a speech, but rather the way it is textually presented by a politician. As for the last case, it could be possible that users might use (low) scores of politicians as an argument against others that support said politicians. Not only is the used data uncertified, we discourage using these results as truth because of the influences discussed in the previous section.

## 10.5 Planning

In the fifth paragraph of chapter six, our planning for this project is shown, supported by a brief description thereof. As can be seen, we initially thought that we needed two weeks for the research phase, as this was also advised by the bachelor thesis guidelines. This however turned out to be almost four weeks instead, which forced us to start later with the remaining phases. The extension was due to several reasons:

- We had to investigate whether describing speech in the form of (some sort of) values was even possible to begin with. For this, we first had to gain knowledge on Natural Language Processing, and how corresponding methods were applied on both speech and written texts.
- We had to determine our approach: which kind of text features did we want to use for the analysis? How are we going to implement the features we selected? Are there any papers or other existing projects that could assist us with the implementation?
- And finally, we had to determine which kind of sources we were able and allowed to use. For the features, think of dictionaries and tool-kits. As input data; documents retrieved via the OData endpoint of the API.

As a result, there was less time left for implementing both backend and frontend. For the backend, this meant that we were unable to implement all features we had defined. This however was also partially due to the frontend; after performing the analysis, we wanted to make sure we could visualize the obtained values. This would also take a considerable amount of time. For the frontend itself, the delay as caused by the extended research phase lead to web application being less interactive. The lack of, for example, the option for users to filter on parties, rather than just on individual members, or on the predefined filters as mentioned previously in section 5.3. Another factor that impacted both backend and frontend was the time it took in order to gain access to the API, as this took (way) longer than expected. The members in our group that received an email with log-in details for the API were unable to access it, and the one member that did not get a response eventually found out she actually could. Also, understanding and processing the data from the API turned out to be quite hard.

## 10.6 Overall performance

As for ourselves, we might have been a little too ambitious regarding everything we initially wanted to implement, both for backend and frontend. During the research phase it was very difficult in deciding what our project goal would be. Since it started out with the idea of exploring and maybe solving the filter bubble phenomenon, and gradually evolved to analysing speech, our focus was not always where it was supposed to be. Therefore we had much trouble in formulating our research questions – they have been changed multiple times, even during implementation. Once we agreed on the direction of our project, the significance of our research became clear and we finally decided to start on implementing. This first meant following tutorials for python, python's NLTK package and Bootstrap. These were all new domains for us, so fully understanding them took longer than desirable.

For the data retrieval we all had to register to get our IP-addresses on a white-list and receive login information for various endpoints. After waiting a week for approval, two of us received confirmation and login data. Unfortunately both their IP-addresses had changed in the mean time. The 3rd group member discovered by chance that her IP-address had been approved (she did not receive any form of notification) meaning we could use the OData endpoint. This form of data representation was also a surprise to us, as stated earlier.

During this project we learned two major things: being decisive and taking human interference into account. The former refers to us defining our problem. On one hand we wanted to actually "work" on something as soon as possible, on the other it still bothered us some things were not clear yet. Things like the text features and the idea of textually analysing speech were fixed elements in our project. But not being able to connect these elements with the actual contribution we would be delivering, kept complicating our process. The second lesson refers to unforeseen obstacles. The first being the API; once we had access the structure of the data was difficult to understand. Once we came to an understanding we encountered human errors.

The second time was when working on the frontend. We wanted to present our results as interactive and appealing as possible. Given the time and our minimal experience in web-development, we decided on using existing implementations of the d3 JavaScript library. Transforming these implementations for our use cases was often frustrating as it was someone else's code. If it did not work as it should we did not know if it was because we did not understand the code or if it was full of bugs or hard-coded data.

The third experience is being human ourselves. Sometimes we overestimated ourselves, creating delays. As said before, we were very ambitious (and maybe still are) and did not stick to reality often. Although the aim of our project was not always clear, the more we approached it, the more we enjoyed it. Yes, we are disappointed we could not deliver what we had in mind. But we are proud in what we did deliver. With every step we made, the contribution we could deliver interested us more and made us believe in its full potential.

# 11 Conclusion and Future Work

## Conclusion

As seen in our Results, we can use predefined text features to describe the language used in the DHoR and make this representable. It gives us insight on how one's language can differentiate and what could be interesting to conclude. Representing the results has been done in several ways, both in our report and in our Data Analysis Tool. During representation it was important to not force any conclusions. Thus we used different means of visualisation and combinations of data. Each visualisation gives room for different conclusions, but as stated in the Discussion, knowledge of how the data was computed is needed to make correct conclusions. Though minimal, the Data Analysis Tool does provide the means for exploring the available results. Meaning a user can, for him-/ herself, decide to what extend languages can be described with text features.

## Future work

For future work, we could implement the missing non-topical text features; doublespeak, etc. We could also introduce the defined topical text features and filter methods, so that we could present a greater variety of features the user can explore. We'd also like to add the function that would allow a user the search for a whole party, rather than a single Representative. Adding Ministers is a similar possibility. In order to gain more knowledge over the DHoR, we could retrieve information for more years, and (if we are able to process the files) create larger time-lines that perhaps would give us new insights.

# References

[1] M. R. Mehl and K. G. Niederhoffer, "Psychological aspects of natural language. use: our words, our selves." *Annual Review of Psychology*, vol. 54, pp. 547–577, 2003.

[2] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," DTIC Document, Tech. Rep., 1975.

[3] R. Prabowo and M. Thelwall, "Sentiment analysis: A combined approach," p. 1, 2009.

[4] G. Katz, N. Ofek, and B. Shapira, "Consent: Context-based sentiment analysis," *Knowledge Based Systems*, vol. 84, pp. 162–178, 2015.

[5] A. Hassan, H. Korashy, and W. Medhat, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, pp. 1093–1113, 2014.

[6] B. Yu, S. Kaufmann, and D. Diermeier, "Classifying party affiliation from political speech," *Journal of Information Technology & Politics*, vol. 5, no. 1, pp. 33–48, 2008.

[7] C. F. Karpowitz and T. Mendelberg, *The Silent Sex Gender: Deliberation, and Institutions*. Princeton University Press, 2014.

[8] J. J. Jones, "Talk "like a man": The linguistic styles of hillary clinton, 1992–2013," *Perspectives on Politics*, vol. 14, no. 3, pp. 625–642, 2016.

[9] "Gender inequality index," 2015. [Online]. Available: http://hdr.undp.org/en/composite/GII

[10] "Women in national parliaments," March 2017. [Online]. Available: http://www.ipu.org/wmn-e/classif.htm

[11] F. Deking, C. Kraaikamp, H. Lopuhaa, and L. Meester, *A Modern Introduction to Probability ans Statistisc*. Delft, The Netherlands: Springer-Verlag London Limited, 2005.

[12] M. Toplak and A. N. Katz, "On the uses of sarcastic irony," *Journal of pragmatics*, vol. 32, no. 10, pp. 1467–1488, 2000.

[13] B. Keysar, "The illusory transparency of intention: Linguistic perspective taking in text," *Cognitive psychology*, vol. 26, no. 2, pp. 165–208, 1994.

[14] V. Hatzivassiloglou and H. Yu, "Toward answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," *EMNLP '03 Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129–136, 2003.

[15] V. Hatzivassiloglou and Y. M. Wiebe, “Exposure to ideologically diverse news and opinion on facebook,” *COLING ’00 Proceedings of the 18th conference on Computational linguistics*, vol. 1, pp. 299–305, 2000.

[16] W. E. Lutz, *Beyond Nineteen Eighty-Four: Doublespeak in a Post-Orwellian Age.* Princeton University Press, 1989.

[17] G. Bohner, “Writing about rape: Use of the passive voice and other distancing text features as an expression of perceived responsibility of the victim,” *British Journal of Social Psychology*, vol. 40, no. 4, pp. 515–529, 2001.

[18] N. M. Henley, M. Miller, and J. A. Beazley, “Syntax, semantics, and sexual violence agency and the passive voice,” *Journal of Language and Social Psychology*, vol. 14, no. 1-2, pp. 60–84, 1995.

[19] E. Tarone, S. Dwyer, S. Gillette, and V. Icke, “On the use of the passive and active voice in astrophysics journal papers: With extensions to other languages and other fields,” *English for specific purposes*, vol. 17, no. 1, pp. 113–132, 1998.

[20] D. R. Olson and N. Filby, “On the comprehension of active and passive sentences,” *Cognitive Psychology*, vol. 3, no. 3, pp. 361–381, 1972.

[21] A. Hunt and B. Wheeler, “Brexit: All you need to know about the uk leaving the eu,” April 2017. [Online]. Available: http://www.bbc.com/news/uk-politics-32810887

[22] “Brexit: Europe stunned by uk leave vote,” June 2016. [Online]. Available: http://www.bbc.com/news/uk-politics-eu-referendum-36616018

[23] L. Wentz, “No brexit-like surprise victory is ahead for trump, says pundit who got brexit right.” [Online]. Available: http://adage.com/article/campaign-trail/trump-pull-a-brexit-surprise-victory/306385/

[24] “Who said brexit was a surprise?” June 2016. [Online]. Available: http://www.economist.com/blogs/graphicdetail/2016/06/polls-versus-prediction-markets

[25] “Official 2016 presidential general election results,” January 2017. [Online]. Available: http://www.fec.gov/pubrec/fe2016/2016presgeresults.pdf

[26] P. Healy and J. W. Peters, “Donald trump’s victory is met with shock across a wide political divide,” November 2016. [Online]. Available: https://www.nytimes.com/2016/11/10/us/politics/donald-trump-election-reaction.html?_r=0

[27] N. Jackson and A. Hooper, “Forecast president senate,” November 2016. [Online]. Available: http://elections.huffingtonpost.com/2016/forecast/president

[28] “General election trump vs clinton. (n.d.).” [Online]. Available: http://www.realclearpolitics.com/epolls/2016/president/us/general_election_trump_vs_clinton-5491.html

[29] E. Bell, “The truth about brexit didn’t stand a chance in the online bubble,” July 2016. [Online]. Available: https://www.theguardian.com/media/2016/jul/03/facebook-bubble-brexit-filter

[30] E. Bakshy, S. Messing, and L. Adamic, “Exposure to ideologically diverse news and opinion on facebook,” *Science*, vol. 348, pp. 1130–1132, 2015.

[31] R. D. Young, “The google “filter bubble” and its problems,” May 2011. [Online]. Available: https://www.searchenginejournal.com/the-google-filter-bubble-and-its-problems/29879/

[32] “Survey: News use across social media platforms 2016,” January-February 2016. [Online]. Available: http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/

[33] C. Doherty, “7 things to know about polarization in america,” June 2014. [Online]. Available: http://www.pewresearch.org/fact-tank/2014/06/12/7-things-to-know-about-polarization-in-america/

[34] L. Bode, E. K. Vraga, P. Borah, and D. V. Shah, “A new space for political behavior: Political social networking and its democratic consequences,” *Journal of Computer-Mediated Communication*, vol. 19, pp. 414—429, 2013.

[35] A. V. Bosch, M. V. Gompel, I. Hendrickx, and K. V. Sloot, *Frog: A Natural Language Processing Suite for Dutch*, 2016.

[36] S. Misailovic and K. Yessenov, “Sentiment analysis of movie review comments,” pp. 1—9, 2009.

[37] E. van Miltenburg, M. Postma, R. Segers, A. Schoen, and P. Vossen, “Open dutch wordnet,” pp. 300–308, 2016.

[38] J. Bauer, S. Bethard, J. Finkel, C. Manning, D. McClosky, and M. Surdeanu, “The stanford corenlp natural language processing toolkit,” 2009.

[39] S. Auer, C. Bizer, S. Hellmann, R. Isele, J. Robert, M. Jakob, A. Jentzsch, P. van Kleef, D. Kontokostas, J. Lehmann, P. Mendes, and M. Morsey, “Dbpedia - a large-scale, multilangual knowledge base extracted from wikipedia,” 2012.

# A Text Features

In this section we present text features that were considered, but not implemented.

### Sarcasm

Sarcasm indicates indirect criticism towards the addressee. In [12] it is shown that sarcasm is, apart from being used as a humorous tool, often used as a means of verbal aggression (mockery, anger-provoking, impoliteness). Sarcasm can also enhance memorability. Detection of sarcasm depends on perspective; knowledge of the topic discussed and the scenario in which it's discussed [13]. Since we will be analyzing text, we wont have intonation or body language indicating sarcasm. We need to take the perspective of the speaker and addressee into account to determine the presence of sarcasm (in our case the whole house or members of the parliament). By using the background of the ones involved we can measure if sarcasm would be detected and/ or if it is used intentionally.

### Subjective/objective

In [14] it is stated that in order to train a Naive Bayes algorithm for sentiment detection, features are included that count the amount of positive and negative words in a sentence. This is because [15] argues that when such words are used in a sentence, it can be an indication that the particular sentence is subjective. This is because opinions are more likely to be expressed through words with a sentimental value rather than ones without.

### Doublespeak

In 1989 Lutz writes the following definition of doublespeak in the book "*Beyond Nineteen Eighty-Four: Doublespeak in a Post-Orwellian Age.*": The word doublespeak combines the meanings of Newspeak and doublethink. Doublespeak is language which pretends to communicate but really does not. It is language which makes the bad seem good, something negative appear positive, something unpleasant appear attractive, or at least tolerable. It is language which avoids or shifts responsibility; language which is at variance with its real and its purported meaning; language which conceals or presents thought. Doublespeak is language which does not extend thought but limits it."[16]. A couple of examples of doublespeak are: "*downsizing*" instead of firing people, *"Take down"* in military language instead of saying killing someone and "*Put to sleep*" instead of euthanize[23].

Doublespeak is widely used in marketing, propaganda and politics throughout the world. We want to measure what the average rate of doublespeak is per politician and how it changes over time and under different debated topics.

[23] http://examples.yourdictionary.com/examples-of-doublespeak.html

By gathering a list of common doublespeak in Dutch or English (and then translate it to Dutch) we can create a measurement of Doublespeak in Dutch parliament.

### Speech rate

By analysing the speech rate, we may find which topics in particular affect a politician.

### Active and passive voice (a supporting feature)

Depending on what the speaker wants to emphasize, he/she will use passive or active voice (sentences). Does he/she want to focus the attention on the actor in the sentence or the one acted upon? In news media, when reporting certain incidents, the passive voice is often chosen to create more acceptance and empathy with the victim(s) [17, 18]. However, when presenting oneself or the group he/she belongs to, the active voice may be chosen to guide the focus of the listeners [19]. This feature holds relevance when we can link a certain motive to it, meaning we will use it as a support feature to enhance others. It is indirectly used in the Feminine to masculine feature, but can also be used to examine how a politician builds sentences when discussing a certain topic (Section 5.2) or the speech's complexity [20].

## B Social media filter bubble

In the past year there have been two moments where political events had results that caused global surprise. The first one, on the 23rd of June 2016, was the Brexit: a plan for Britain to leave the EU that was officially put into practice when a majority of 51.9% of the British people voted to leave during the British referendum [21]. Not only were civilians from in- and outside Britain surprised with the outcome [22], but also professionals [23] and polls [24] found themselves in the dark.

Something similar happened when Donald Trump got elected President of the United States of America [25]. The outcome of this presidential election caused an even greater shock worldwide. Again among civilians [26], professionals [27] and polls [28], the difference between predictions and the eventual outcome was great. The interesting part is that for these outcomes to happen there had to be big groups of voters who seemed “hidden”. These groups determined the results of the elections, yet voters on the other side had not imagined their opponents were so numerous.

Many questions arise from these outcomes. How did this happen? How is it possible that no one saw it coming? How can there be such isolated groups of people barely noticing each other's existence? The answer lies within technology and the Internet. Apparently voters, professionals and even poll conductors reside in an “online bubble” [29]. The feed Facebook users see when they log into their personal account is made out of content presented to them via Facebook friends and liked pages, and is filtered out by algorithms that only try to show content that users will enjoy. This content is biased since it is created or shared by users with similar views and only partially visible to the users because of the Facebook filtering algorithm [30]. Not only Facebook has been reported to have a so-called ”filter bubble” problem: almost all information distribution websites make use of such filtering algorithms and personalized recommendation systems, such as: social media platforms (besides Facebook), Google[31] and newspapers.

**About 6-in-10 Americans get news from social media**

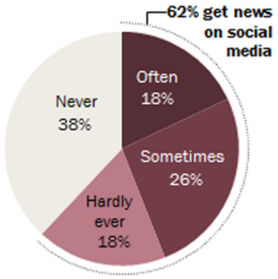


Figure 31: % of U.S. adults who get news on a social networking site [32]

**Growth in use of social media for news**

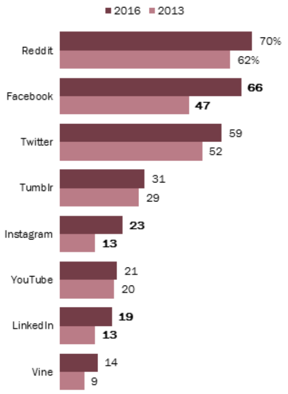


Figure 32: % of users of each social networking site who get news there [32]

**Most social media news consumers only get news on one site**

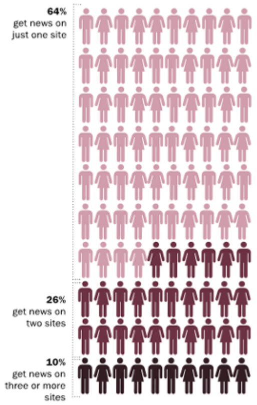


Figure 33: % of news users of at least one social media site who [32]

The fact that numerous users on the Internet encounter biased content on websites on a daily basis would not form a significant problem if this content would only consist of a small part of their observed news. However, studies show that adult users in the US, for example, 62% of the social media users, get news from social media as is visible in Figure 31. Figure 32 shows that currently more than 50% of the users of the three biggest social media platforms, respectively Reddit, Facebook and Twitter, get news from those platforms. Also figure 32 shows that this number has increased between 2013 and 2016 (the statistically most significant differences are shown in **bold**). Also out of the users who use at least one social media site, 64% get news from just one site, which is shown in figure 33. Overall, these three figures above show that social media sites play a role in news gathering of its users, that the influence of social media on news gain has increased over the past years and finally a great percentage of social media users only have few sources for their news.

The US has become more politically polarized over the past 13 years, [33] and social media can be pointed out as one of the causes [34]. Peoples' clicks and likes combined with social media filter algorithms create separate networks of users with similar views.

# C Toolkits and Knowledge Bases

## C.1 Toolkits

As stated earlier, we want to make use of sentiment analysis in order to profile and characterize our politicians. For this type of analysis, there exist so-called toolkits. A toolkit[24] is a collection of libraries that provide the user with a greater variety of possible solutions, as compared to the options the developer's environment initially supports. So, in order to avoid starting from scratch and reinventing the wheel, we sought a toolkit that enabled us to achieve an implementation that meets our requirements. In the following section, we provide a description about current toolkits that could contribute to such an implementation, discuss their pros and cons, and in the end justify our eventual choice. We did however also investigate toolkits that do not provide a sentiment analyser, but do actually offer components that could enhance the overall performance; a better sentence analyzer compared to our eventual choice.

### C.1.1 Current toolkits

***Frog***[25] is powerful tool when it comes to sentence segmentation [35]. It is able to determine what grammatical type of words the sentence contains, providing a speech tag for each word the algorithm comes across. Here, it shows how 'certain' the assignment is by adding a confidence number. Frog thus also introduces the option to tokenize and lemmatize its input text. To us, Frog distinguished itself by the fact that it is completely based on the Dutch language. It uses corpora that contain hundreds of millions of words, and thus ensures any text written in Dutch can be thoroughly analyzed. The downside however, is that Frog is a finished product: extending it with a sentiment analyzer might prove to be extremely difficult. This is the main reason we decided not to initially use this toolkit.

***Natural Language Toolkit(NLTK)***[26] is a python based language processor composed of many libraries and programs that provide tools for sentence and sen-

[24] http://stackoverflow.com/questions/5453011/api-vs-toolkit-vs-framework-vs-library

[25] https://languagemachines.github.io/frog/

[26] http://www.nltk.org/

timent analysis [36]. NLTK also allows easy access to more than fifty corpora and other databases focused on the classification of different languages. Furthermore, there is a lot of documentation on how the NLTK packages can be properly used, and how these packages can be combined in order to build a complete sentiment analyser. Regarding our requirements, the downside that NLTK has is that it does not initially support the Dutch language through its largest lexicon; WordNet, which revolves around the English language. A Dutch version[27] [37] of WordNet however does exist, so it should be possible to integrate this in the toolkit.

***Stanford NLP toolkit***[28] is also a very powerful tool that covers a lot of our requirements; speech tagging, named entity recognizing etc. [38]. Also, the tool is written in Java, which is the programming language we are most familiar with. The downside is that this toolkit does not seem to support anything written in Dutch. (We came across a sub-corpora download that enabled speech tagging in Dutch, but this library only contained around 150,000 words - as compared to 30 million for US English, so this seemed insufficient.)

### C.1.2 Choice Justification

After considering the possible options each toolkit could provide, we decided to use NLTK. As stated, the NLTK comes with a lot of methods that could help realizing a desired implementation, also regarding both accuracy and efficiency. Furthermore, the existing documents clearly describe the necessary steps that are needed to be taken in order to achieve a basic sentiment analyzer. Since we are still in the very first phase of the project, it is of course possible that we will come across tools that might perform certain tasks better. Because of this, other toolkits, either mentioned here or new ones, could be used as well in the (near) future.

## C.2 Knowledge bases

In the previous section we discussed some toolkits that could help achieve a sufficient or improved implementation, and in the end concluded that NLTK seemed the best option to start off with. As mentioned, this toolkit does not initially provide a very large database regarding the semantics of the Dutch language, so an additional download might be necessary. In this section, we will discuss such external semantic databases, and their potential contribution.

[27] http://wordpress.let.vupr.nl/odwn/

[28] https://stanfordnlp.github.io/CoreNLP/

### C.2.1 Yago

Yago[29] or Yet Another Great Ontology, is a semantic database that combines the knowledge of other sources, such as Wikipedia, Geonames[30] and WordNet into one massive suite. This database holds a 95% confirmed accuracy, as its suite has been manually tested. Furthermore, Yago also maintains the structure of WordNet, which, as stated in the previous section, is already integrated in NLTK. Therefore, it should be possible to easily swap the initial version of WordNet for this expanded and improved database. In our case, the greatest property of this knowledge base however is that it supports the Dutch language, as it extracts Dutch Wikipedia pages in order to construct its suite. If we would thus be able to use Yago instead of NLTK's version of WordNet, this would make for a huge improvement. An extra aspect of Yago we might use is its option to cluster words based on thematic domains. That is, if it is supports themes related to the Dutch parliament.

### C.2.2 DBpedia

Another knowledge base we looked at is a community powered project called DBpedia[31] (where DB stands for database). DBpedia aims to extract structures from Wikipedia pages and stores them into its own database. Users can then define their own specific query's and use these to retrieve desired information from the DBpedia suite. As stated in [39] this knowledge base can also be used for applications in the domains of Named Entity Recognition and topic detection. Like Yago, DBpedia supports the Dutch language because of its information retrieval through Wikipedia.

[39] also includes a section comparing Yago and DBpedia. Here, it is mentioned that Yago achieves higher precision and more consistent knowledge because it focuses on extracting a smaller number of relations, as compared to DBpedia. Also, since Yago makes use of WordNet rather than being manually powered, it contains more classes than DBpedia. For named entity databases, we thus preferred Yago.

### C.2.3 SentiStrength

SentiStrength[32] is an application that allows the user to perform sentiment analysis over written texts. The algorithm assigns two values to each word in the text, the first varying between -1 and -5 to indicate negativity, and the second varying between 1 and 5 to indicate positivity. As a result, the text, or parts thereof, can be labeled with an overall mean 'score'. Although given the option on its website, SentiStrength was not initially able to analyse an example sentence written in Dutch, as none of the words were assigned a value. We added

[29] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[30] http://www.geonames.org

[31] http://wiki.dbpedia.org/about

[32] http://sentistrength.wlv.ac.uk/

an external list of Dutch words with matching sentiment score from an open source project on GitHub[33] and started the analyser again. This however still resulted in an obvious lack of basic sentiment words such as ’boos’ (meaning angry in Dutch), so we deemed this analyser to be insufficient. Furthermore, it is also explicitly stated on its website that SentiStrength could replicate human accuracy when it comes to short social texts, written in English, *except* political texts. This was the main reason we decided not to use this application.

### C.2.4 Pattern

Another module that contains a sentiment database we looked at is called Pattern[34]. Pattern has tools for both data mining and natural language processing, including a sentiment analyser. Like SentiStrength, it returns a pair of values to assign an overall score to a piece of text. In Pattern however, the first value varies between -1.0 and 1.0 indicating polarity, while the second value, varying between 0.0 and 1.0, indicates subjectivity. The option to also get an estimate of a text’s subjectivity was one we deemed to be very interesting. Finally, Pattern supports Dutch. Overall, this module seems to meet our requirements, so as for sentiment databases, we preferred Pattern.

## D Performance Testing

Testing the quality of our software is an obvious necessity throughout the entire development process. During each phase, we should test its stability and determine whether its generated output is correct or not. Since it is not exactly known what our eventual results will look like however, numerous testing methods cannot be fully applied to our particular application.

For example, Black-box testing would require a set of correct results, which could then be used to validate the output of our application. Although we will be able to point out extreme cases manually, this output will always vary in a certain range, and in our case, the output will most definitely not be the same when looking at multiple politicians. When considering the Flesch-Kincaid score for example, we expect it to be between 6 and 12 for any politician. We deem these to be valid results. If any outliers are detected, we can trace back their origin and determine if this is due to an error in our software or not. The Black-box testing sub-technique that is most similar to this approach is focussing on boundary values; define a range of values (a minimum and maximum), and verify if our results can be mapped into it.

White-box testing however can be applied in many instances. Its testing techniques, control-flow and data-flow testing, allow us to improve the structure and efficiency for each phase during the development process. In order to perform control-flow testing, we will define test cases that cover as many - preferably all - of the statements within each component of the application; a

[33] https://github.com/felipebravom/StaticTwitterSent/tree/master/extra/SentiStrength/dutch

[34] http://www.clips.ua.ac.be/pages/pattern-nl

set of paths. These paths could for example help us detect unreachable code. For data-flow testing, we could define test cases that cover all variables per component of our application, and determine if they were correctly changed, initialized, etc. Again, as stated earlier, it is not possible to fully ensure correctness of the generated output, which in turn might leave some smaller errors in our code to be unnoticed. This also impedes the integration of Unit Tests; even on component level, correct results may be impossible to be perfectly defined.

As for Performance testing, we can insert a very great amount of data - an amount that could however be a possible situation in extreme cases - and check if the algorithm does not get overloaded or that information will be lost during the analysis. In our case, we want to get as much input as we can; all documented speeches during debates per politician - so this testing method is of great importance.

We will also focus on Regression testing; we can run an updated or changed component again and verify if its output didn't change compared to its previous state. Here, we can determine whether the mutation had a negative (or positive) effect on the application.

Finally, we will test the users' experience of the application by means of their interaction with the interface provided by the corresponding website. In this Acceptance testing phase, we will first determine for ourselves whether we deem the layout, functionality, readability etc., to be sufficient or not, and in the following stage, let a group of users test our application and process their feedback.

In general, these different types of testing ensure that the application will be thoroughly tested and maintained throughout the entire development process, and that its final version will meet as many of our proposed requirements.

# E Success Criteria

The requirements for our research and consequential product are stated in Section 6, mainly in tables 5 and 6. In means of deliverables we would like to implement at least the requirements stated in the ”Should have” sections of the tables. Our process, and therefore our implementations, should be well documented in a report which will be delivered as well. In some way the report itself is the most important as, at the beginning of our project, we do not know what (meaningful) results will be discovered during different phases. Or, if they will be found at all. Thus we should be able to provide support as to why and how this outcome could occur. In general, our findings, whether eventual or partial, should be accompanied by explanation through analysis, and, if possible, earlier results.

## E.1 Analysis with text features

### E.1.1 Text features

After relevant text features have been chosen, their contribution to our research (questions) should be shown. Does this text feature, on its own or in combination with other text features, enhance our results? Does this text feature actually give more insight into a politician's speech pattern? If these questions can be answered positively, success is met. If this is not the case, we would want to document it as well to show why it was not included.

### E.1.2 Topics and temporal influences

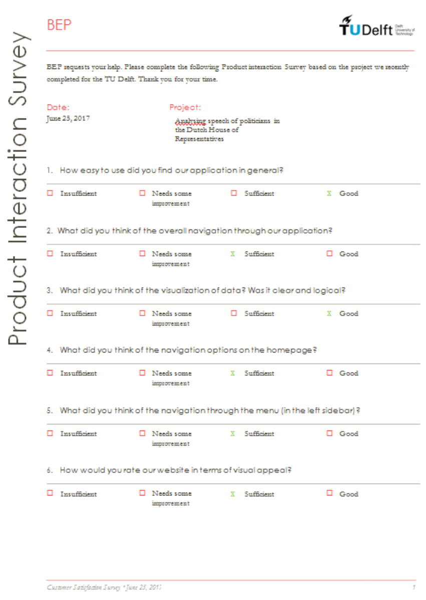
We need to find topics and different points in time which have an impact on a politician's speech pattern; cause a variance. We expect that, for each party in particular, topics which are of great(er) importance are likely to cause such a variance. For temporal influences we are not bounded by the party, as important periods or events apply to the whole house. Here, success criteria will be met if we succeed in finding cases that induce variances. However, if (expected) variances are not introduced, we will provide an explanation. If this is deemed to be impossible, a reasoning as to why this is the case will suffice.

## E.2 Visualisation

When it comes to visualization, it is crucial that we present our findings in a clear way to the user. The latter should, through quite straight-forward steps, be able to request information without any confusion being introduced. Here, we must focus on structuring the website in such a way that it provides easy access to its tools and data. Naturally, it is also very important the user understands the visualized results of his or her query. Therefore, we have to determine which visualization method(s); graphs, diagrams, box-plots, etc., best suit the type of information to be presented. The combination of these two aspects describes the success criteria for this particular component; namely, the application should be easy to use, and the overall goal we want to achieve - making our research accessible and therefore giving a new form of insight in the House of Representatives - should become clear.

# F Usertest surveys

## User tests surveys



Product Interaction Survey

BEP

BEP requests your help. Please complete the following Product interaction Survey based on the project we recently completed for the TU Delft. Thank you for your time.

Date:
June 25, 2017

Project:
Analysing speech of politicians in the Dutch House of Representatives

1. How easy to use did you find our application in general?

☐ Insufficient ☐ Needs some improvement ☐ Sufficient X Good

2. What did you think of the overall navigation through our application?

☐ Insufficient ☐ Needs some improvement X Sufficient ☐ Good

3. What did you think of the visualization of data? Was it clear and logical?

☐ Insufficient ☐ Needs some improvement ☐ Sufficient X Good

4. What did you think of the navigation options on the homepage?

☐ Insufficient ☐ Needs some improvement X Sufficient ☐ Good

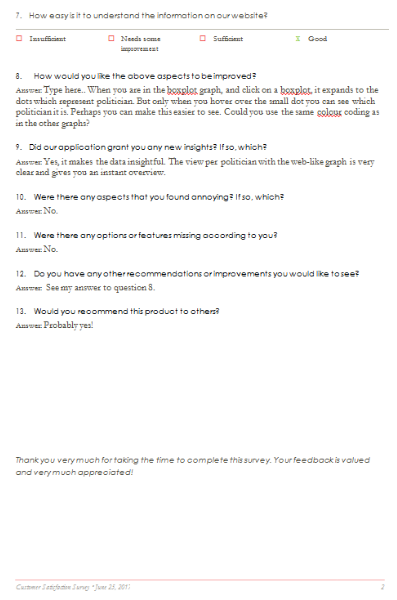
5. What did you think of the navigation through the menu (in the left sidebar)?

☐ Insufficient ☐ Needs some improvement X Sufficient ☐ Good

6. How would you rate our website in terms of visual appeal?

☐ Insufficient ☐ Needs some improvement X Sufficient ☐ Good

Customer Satisfaction Survey • June 25, 2017 1



7. How easy is it to understand the information on our website?

☐ Insufficient ☐ Needs some improvement ☐ Sufficient X Good

8. How would you like the above aspects to be improved?

Answer: Type here.. When you are in the boxplot graph, and click on a boxplot, it expands to the dots which represent politician. But only when you hover over the small dot you can see which politician it is. Perhaps you can make this easier to see. Could you use the same colour coding as in the other graphs?

9. Did our application grant you any new insights? If so, which?

Answer: Yes, it makes the data insightful. The view per politician with the web-like graph is very clear and gives you an instant overview.

10. Were there any aspects that you found annoying? If so, which?

Answer: No.

11. Were there any options or features missing according to you?

Answer: No.

12. Do you have any other recommendations or improvements you would like to see?

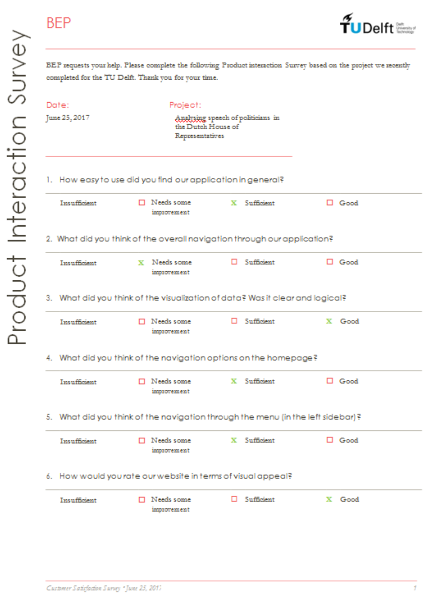
Answer: See my answer to question 8.

13. Would you recommend this product to others?

Answer: Probably yes!

*Thank you very much for taking the time to complete this survey. Your feedback is valued and very much appreciated!*

Customer Satisfaction Survey • June 25, 2017 2



Product Interaction Survey

BEP

BEP requests your help. Please complete the following Product interaction Survey based on the project we recently completed for the TU Delft. Thank you for your time.

Date:
June 25, 2017

Project:
Analysing speech of politicians in the Dutch House of Representatives

1. How easy to use did you find our application in general?

Insufficient ☐ Needs some improvement X Sufficient ☐ Good

2. What did you think of the overall navigation through our application?

Insufficient X Needs some improvement ☐ Sufficient ☐ Good

3. What did you think of the visualization of data? Was it clear and logical?

Insufficient ☐ Needs some improvement ☐ Sufficient X Good

4. What did you think of the navigation options on the homepage?

Insufficient ☐ Needs some improvement X Sufficient ☐ Good

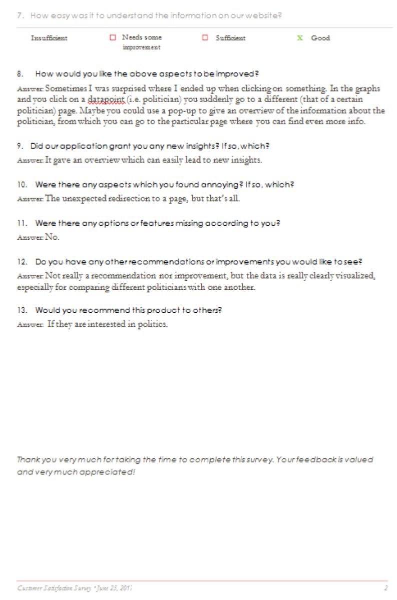
5. What did you think of the navigation through the menu (in the left sidebar)?

Insufficient ☐ Needs some improvement X Sufficient ☐ Good

6. How would you rate our website in terms of visual appeal?

Insufficient ☐ Needs some improvement ☐ Sufficient X Good

Customer Satisfaction Survey • June 25, 2017 1



7. How easy was it to understand the information on our website?

Insufficient ☐ Needs some improvement ☐ Sufficient X Good

8. How would you like the above aspects to be improved?

Answer: Sometimes I was surprised where I ended up when clicking on something. In the graphs and you click on a datapoint (i.e. politician) you suddenly go to a different (that of a certain politician) page. Maybe you could use a pop-up to give an overview of the information about the politician, from which you can go to the particular page where you can find even more info.

9. Did our application grant you any new insights? If so, which?

Answer: It gave an overview which can easily lead to new insights.

10. Were there any aspects which you found annoying? If so, which?

Answer: The unexpected redirection to a page, but that's all.

11. Were there any options or features missing according to you?

Answer: No.

12. Do you have any other recommendations or improvements you would like to see?

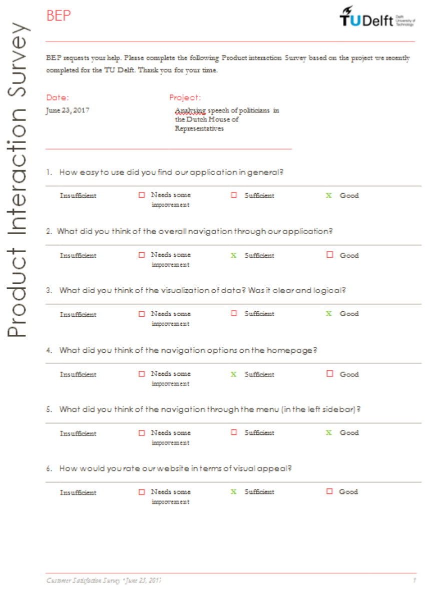
Answer: Not really a recommendation nor improvement, but the data is really clearly visualized, especially for comparing different politicians with one another.

13. Would you recommend this product to others?

Answer: If they are interested in politics.

*Thank you very much for taking the time to complete this survey. Your feedback is valued and very much appreciated!*

Customer Satisfaction Survey • June 25, 2017 2



Product Interaction Survey

BEP

BEP requests your help. Please complete the following Product interaction Survey based on the project we recently completed for the TU Delft. Thank you for your time.

Date:
June 23, 2017

Project:
Analysing speech of politicians in the Dutch House of Representatives

1. How easy to use did you find our application in general?

Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

2. What did you think of the overall navigation through our application?

Insufficient ☐ Needs some improvement ☒ Sufficient ☐ Good

3. What did you think of the visualization of data? Was it clear and logical?

Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

4. What did you think of the navigation options on the homepage?

Insufficient ☐ Needs some improvement ☒ Sufficient ☐ Good

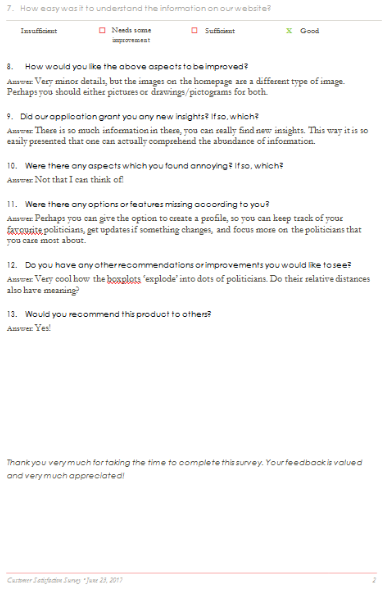
5. What did you think of the navigation through the menu (in the left sidebar)?

Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

6. How would you rate our website in terms of visual appeal?

Insufficient ☐ Needs some improvement ☒ Sufficient ☐ Good

Customer Satisfaction Survey • June 23, 2017 1



7. How easy was it to understand the information on our website?

Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

8. How would you like the above aspects to be improved?

Answer: Very minor details, but the images on the homepage are a different type of image. Perhaps you should either pictures or drawings/pictograms for both.

9. Did our application grant you any new insights? If so, which?

Answer: There is so much information in there, you can really find new insights. This way it is so easily presented that one can actually comprehend the abundance of information.

10. Were there any aspects which you found annoying? If so, which?

Answer: Not that I can think of!

11. Were there any options or features missing according to you?

Answer: Perhaps you can give the option to create a profile, so you can keep track of your favourite politicians, get updates if something changes, and focus more on the politicians that you care most about.

12. Do you have any other recommendations or improvements you would like to see?

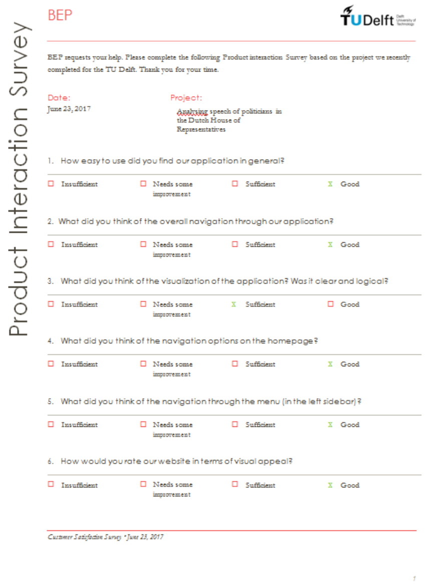
Answer: Very cool how the boxplots 'explode' into dots of politicians. Do their relative distances also have meaning?

13. Would you recommend this product to others?

Answer: Yes!

*Thank you very much for taking the time to complete this survey. Your feedback is valued and very much appreciated!*

Customer Satisfaction Survey • June 23, 2017 2



Product Interaction Survey

BEP

BEP requests your help. Please complete the following Product interaction Survey based on the project we recently completed for the TU Delft. Thank you for your time.

Date:
June 23, 2017

Project:
Analysing speech of politicians in the Dutch House of Representatives

1. How easy to use did you find our application in general?

☐ Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

2. What did you think of the overall navigation through our application?

☐ Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

3. What did you think of the visualization of the application? Was it clear and logical?

☐ Insufficient ☐ Needs some improvement ☒ Sufficient ☐ Good

4. What did you think of the navigation options on the homepage?

☐ Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

5. What did you think of the navigation through the menu (in the left sidebar)?

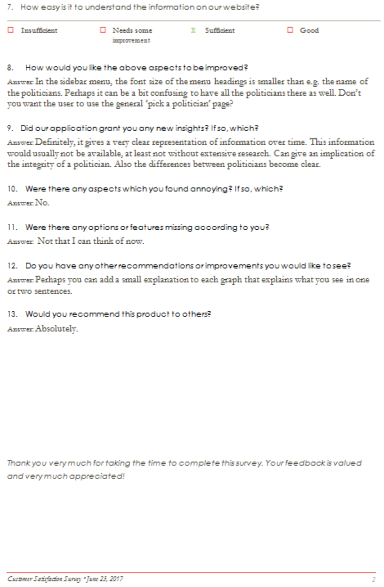
☐ Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

6. How would you rate our website in terms of visual appeal?

☐ Insufficient ☐ Needs some improvement ☐ Sufficient ☒ Good

Customer Satisfaction Survey • June 23, 2017

1



7. How easy is it to understand the information on our website?

☐ Insufficient ☐ Needs some improvement ☒ Sufficient ☐ Good

8. How would you like the above aspects to be improved?

Answer: In the sidebar menu, the font size of the menu headings is smaller than e.g. the name of the politicians. Perhaps it can be a bit confusing to have all the politicians there as well. Don't you want the user to use the general 'pick a politician' page?

9. Did our application grant you any new insights? If so, which?

Answer: Definitely, it gives a very clear representation of information over time. This information would usually not be available, at least not without extensive research. Can give an implication of the integrity of a politician. Also the differences between politicians become clear.

10. Were there any aspects which you found annoying? If so, which?

Answer: No.

11. Were there any options or features missing according to you?

Answer: Not that I can think of now.

12. Do you have any other recommendations or improvements you would like to see?

Answer: Perhaps you can add a small explanation to each graph that explains what you see in one or two sentences.

13. Would you recommend this product to others?

Answer: Absolutely.

*Thank you very much for taking the time to complete this survey. Your feedback is valued and very much appreciated!*

Customer Satisfaction Survey • June 23, 2017 2

# G Research report

# Midterm Report

BEP - Profiling our parliament's politicians

Rebecca Glans, Mila Hendrikse, Martin Koole

May 28, 2017

# Contents

# 1 Summary

The purpose of this project is to analyse textual data from debates of the Dutch House of Representatives in order to profile the language of politicians and visualise patterns in their speech.

The project was inspired by the current social media filter bubble. The social media filter bubble is the phenomenon where online social media users get separated from online content that opposes the users' viewpoints. This isolation is due to and strengthened by personal online behavior (clicks, likes, posts, search history and more) in combination with personalised website filtering algorithms. As this phenomenon expands over time, it enhances the political polarisation within a society; people tend to refuse being positive about politicians they (completely) disagree with. In this online "bubble" a user is only exposed to a tiny part of all information about politicians' remarks, decisions and activities and derives his/her opinions and conclusions from this biased and limited information. It is questionable if the root of these isolated groups of political followers lies purely with their social and online setting which offers them limited information, or if the language used by different politicians has such a great influence on followers that they could never remotely appeal to certain groups of civilians. Therefore we will not look at limited and biased content, but to the actual words spoken in actual debates in the Dutch House of Representatives. We will we examine (computable) textual aspects which could play a part in the impression politicians can leave on their listeners in order to analyse speech patterns of politicians and how they vary under different circumstances. With these retrieved patterns we will be able to analyse how similar/different the language of different politicians actually is and cluster their language based on various criteria. We aim to give users the ability to draw their own conclusions based on the visualization(s) of these clusters, as this application uses completely unbiased, raw data, ensuring the absence of filter bubbles.

# 2 Problem definition and analysis

## 2.1 Problem definition

In the past year there have been two moments where political events had results that caused global surprise. The first one, on the 23rd of June 2016, was the Brexit: a plan for Britain to leave the EU that was officially put into practice when a majority of 51.9% of the British people voted to leave during the British referendum [1]. Not only were civilians from in- and outside Britain surprised with the outcome [2], but also professionals [3] and polls [4] were finding themselves in the dark.

Something similar happened when Donald Trump got elected President of the United Stated of America [5]. The outcome of this presidential election caused an even greater shock worldwide. Again among civilians [6], professionals [7] and polls [8], the difference between predictions and the eventual outcome was great. The interesting part is that for these outcomes to happen there had to be big groups of voters who seemed “hidden”. These groups determined the results of the elections, yet voters on the other side had not imagined their opponents were so numerous.

Many questions arise from these outcomes. How did this happen? How is it possible that no one saw it coming? How can there be such isolated groups of people barely noticing each other’s existence? The answer lies within technology and the Internet. Apparently voters, professionals and even poll conductors reside in an “online bubble” [9]. The feed Facebook users see when they log into their personal account is made out of content presented to them via Facebook friends and liked pages, and is filtered out by algorithms that only try to show content that users will enjoy. This content is biased since it is created or shared by users with similar views and only partially visible to the users because of the Facebook filtering algorithm [10]. Not only Facebook has been reported to have a so-called ”filter bubble” problem: almost all information distribution websites make use of such filtering algorithms and personalized recommendation systems, such as: social media platforms (besides Facebook), Google[11] and newspapers.

**About 6-in-10 Americans get news from social media**

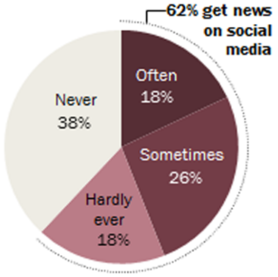


Figure 1: % of U.S. adults who get news on a social networking site [12]

**Growth in use of social media for news**

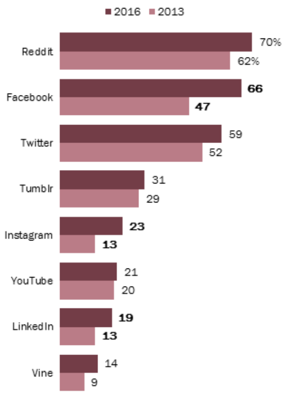


Figure 2: % of users of each social networking site who get news there [12]

**Most social media news consumers only get news on one site**

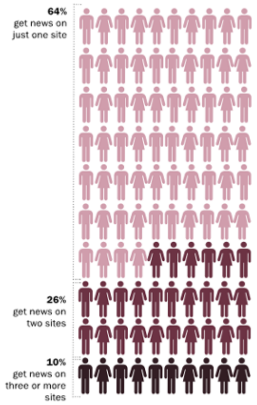


Figure 3: % of news users of at least one social media site who [12]

Users on the Internet who encounter biased content on websites on a daily basis wouldn't form a significant problem if this content would only consist of a small part of their observed news. However, studies show that adult users in the US, for example, 62% of the social media users, get news from social media as is visible in Figure 1. Figure 2 shows that currently more than 50% of the users of the three biggest social media platforms, respectively Reddit, Facebook and Twitter, get news from those platforms. Also figure 2 shows that this number has increased between 2013 and 2016 (the statistically most significant differences are shown in **bold**). Also out of the users who use at least one social media site, 64% get news from just one site, which is shown in figure 3. Overall, these three figures above show that social media sites play a role in news gathering of its users, that the influence of social media on news gain has increased over the past years and finally a great percentage of social media users only have few sources for their news.

The US has become more politically polarized over the past 13 years, [13] and social media can be pointed out as one of the causes [14]. So apparently peoples' clicks and likes combined with social media filter algorithms create separate networks of users with similar views. Imagine we would not have social media filter algorithms and we would only look at raw political data; would we divide ourselves in the same ideological groups? In our project, the debates in the House of Representatives are used as input data. Are there patterns in political speech that would suggest ideological subgroups? That's what our project is about.

## 2.2 Client and project coach

For our Bachelor End Project (BEP) at the Technical University of Delft (TU Delft) we will perform a project with the goal of exploring these patterns in the spoken and written text by politicians. Our client will be dr. Cynthia Liem [1], Assistant Professor at the Multimedia Computing Group at the TU Delft and our project coach is dr. ir. Alessandro Bozzon[2], Assistant Professor with the Web Information Systems group at the TU Delft.

[1] http://mmc.tudelft.nl/users/cynthia-liem

[2] http://www.wis.ewi.tudelft.nl/bozzon/

# 3 Profiling political speech

In order to analyse the way politicians speak, we examine (computable) textual aspects which could play a part in the impression politicians can leave on their listeners. Not only the content of a conversation, a discussion or (political) proposal influences this impression. Also choice of words and sentences of a speaker influences the way a message is transmitted and perceived[15]. We call these textual aspects 'text features', and we divide them into two groups: non-topical and topical test features. We aim to represent the language used in the House of Representatives via these text features. We will not look at what is said, but how it is said textually. As shown in Section 3.1 this will include, among others, a sentiment analysis and an analysis of the complexity of the speech.

With these defined text features which represent speech (in text form), we can analyse and measure how speech of politicians varies among each other or under different circumstances. In Sections 3.2, 3.3 we introduce three variables with whom we define these different circumstances.

## 3.1 Non-topical text features

Intelligent use of language can help politicians make their statements clear, make listeners memorize what is said easier, persuade their colleagues into considering a topic from a different perspective, emphasize their powerful position and much more. We try to turn these influential linguistic aspects into numbers with the text features.

Also we will calculate the average of feature values per politician, which we call the baseline of the politician.

In this section we describe all the text features to be used. We explain their relevance and (in time) how it is computed.

- Complexity
- Doublespeak
- Sentiment Analysis
- Feminine to masculine speech style ratio
- Synonyms (a supporting feature)

### Complexity

How difficult is it to understand a politician? Does he use simple (short) words or words containing many syllables? It can indicate what the speaker expects his audience to know about the topic or, on the other hand, how well-informed the speaker is. Using the Flesch-Kincaid Reading Test [16] we can determine the difficulty of a given text based on the amount of syllables in each word.

### Sentiment Analysis

Different types of texts can be categorized in two main sentimental categories, namely positive and negative [17]. Of course, it is possible to assign a value that lies somewhere between these; defining a scale that allows intermediate values. This can thus also be done on mere word or sentence level [18]. In general this means it is possible to indicate per (piece of) text whether it is linked to positive or negative emotional value. Furthermore, sentiment analysis can be used to analyse political debates[19].

### Doublespeak

In 1989 Lutz writes the following definition of doublespeak in the book *"Beyond Nineteen Eighty-Four: Doublespeak in a Post-Orwellian Age."*: "The word doublespeak combines the meanings of Newspeak and doublethink. Doublespeak is language which pretends to communicate but really does not. It is language which makes the bad seem good, something negative appear positive, something unpleasant appear attractive, or at least tolerable. It is language which avoids or shifts responsibility; language which is at variance with its real and its purported meaning; language which conceals or presents thought. Doublespeak is language which does not extend thought but limits it."[20]. A couple of examples of double speak are: *"downsizing"* instead of firing people, *"Take down"* in military language instead of saying killing someone and *"Put to sleep"* instead of euthanize[3].

Doublespeak is widely used in marketing, propaganda and politics throughout the world. We want to measure what the average rate of doublespeak is per politician and how it changes over time and under different debated topics.

By gathering a list of common double speak in Dutch or English (and then translate it to Dutch) we can create a measurement of Double Speak in Dutch parliament.

### Feminine to masculine speech style ratio

Politics is mostly a male-dominated area. While women are entering decision-making bodies more and more each year, they still form a minority. Because of this, they tend to conform to the style of communication typical for this area that, as a result of the majority of the members being male, has a tendency towards a masculine style of speech [21]. What is interesting about this text feature is that the gender indicating speech of a politician (or human in general) is not fixed; it can be deliberately or even unintentionally fluctuate over time or even between different speeches and debates of the same politician. Take for example United States politician Hilary Clinton. A research about the linguistic style of Hilary Clinton on data from interviews and candidate debates dating between 1992 and 2013 shows that Clinton's linguistic style fluctuates between masculinity and femininity over the years, depending on the different roles she

[3] http://examples.yourdictionary.com/examples-of-doublespeak.html

played over the years (some years it was a more supportive role, others one of pure leadership and) [22].

The Netherlands scores really well on the global Gender Inequality Index (GII)[4], ranking 3rd best place in 2015 according to the United Nations Development Programme[23] while the USA for example ranked 43. The Dutch parliament also scored high globally on percentage of women in parliament on the 1th of March 2017 (so the ratio of the parliament before the most recent elections) with a score of 34.7% ranking 21th place in a report made by the Inter-Parliamentary Union [24]. Because of this relatively powerful ranking of the Netherlands compared to other countries we include a text feature about gender indicative speech. We believe it is interesting how feminine/masculine our House of Representatives speaks and if this speech changes over time and to what extend it differs between female and male politicians.

The ratio of feminine to masculine speech style can be easily calculated by counting feminine and masculine linguistic markers in text and devide them: feminine/masculin[22].

### Synonyms (a supporting feature)

Analysing which words are used can also give an insight in how politicians (wish to) communicate. When engaging in a topic, does one choose a more **modern** or **traditional** version of a word? Does one introduce a self made word to emphasize his opinion or use slang to add humour or show involvement? This feature is also contained in the features above, which is why it is also a supporting feature. For instance, in [25] it is shown that synonyms can carry **sentiment** in certain contexts; instead of using the word abortion, medical terms are used when in a pro-choice position.

### Active and passive voice (a supporting feature)

Depending on what the speaker wants to emphasize, he/ she will use passive or active voice (sentences). Does he/ she wants to focus the attention on the actor in the sentence or the one acted upon? In news media, when reporting certain incidents, the passive voice is often chosen to create more acceptance and empathy with the victim(s) [26, 27]. However, when presenting oneself or the group he/ she belongs to, the active voice may be chosen to guide the focus of the listeners [28]. This feature holds relevance when we can link a certain motive to it, meaning we will use it as a support feature to enhance others. It is indirectly used in the Feminine to masculine feature, but can also be used to examine how a politician builds sentences when discussing a certain topic (Section 3.2) or the speech's complexity [29].

[4] http://hdr.undp.org/en/content/gender-inequality-index-gii

## 3.2 Topical Text features

The topic of a text is in fact a text feature. Thus we will use this text feature in two ways. We will use it to characterize a politician's speech pattern, as the text features in section 3.1. Which topics does he/ she address the most? Is this finding expected when looking at the committee and/ or party this politician belongs to? Our second use will be to explore the politician's speech pattern as described below.

### Campaign topics

During election campaigns each party announces which current issues it wants to address and how it wants to tackle them. These issues can vary from economical to immigration topics and many more. When a party makes it into parliament it is expected of the ministers to take action and work on the issues that were campaigned on in order to satisfy its voters and gain trust We want to see if these specific party topics, because they are of greater political importance to the party than others, have in influence on the speech of party members compared to topics that weren't mentioned in the party campaign.

## 3.3 Temporal influences

Can we analyse how the speech pattern of a politician varies during his/her term? Many things could influence how a politicians speaks; experience, the topic discussed, recent events. Next to giving an overall speech pattern, we would like to show how this speech pattern was formed or influenced. Our first approach is using the topic discussed (3.2). In this section we would like to give two approaches that use different events or periods during a politicians term to determine their influence.

### The political agenda

The date of a debate could change the politicians speech pattern; some politicians for example are known for making more populist statements when the elections are approaching. By taking different (major) events on the political agenda we can analyse how the speech patterns vary.

### Unforeseen important events

The attitude and speech pattern of politicians can vary when news emerges that causes immediate response from- and debates in the House of Representatives. By looking at so-called emergency debates in the House of Representatives we can analyse how the values of the text features of politicians vary under circumstances that urges them to think and debate with little preparation time about stressing topics.

# 4 Research question

We use the text-features described in Section 3 to visualize the text used during debates.

## Main question

When extracting and visualizing text features from political language used in the Dutch House of Representatives, what (individual) speech patterns can be discovered?

## Sub questions

1. How is a politician's speech pattern characterised by text features?
2. How is a politician's speech pattern (based on the text features) influenced by:
   - the topic discussed
   - major (planned) events on the political agenda
   - unforeseen events of political importance
3. What clusters of politicians can we distinguish when visualizing their speech patterns?

# 5 Available Data

Every (plenary) meeting in the House of Representatives is documented by the ”Dienst Verslag en Redactie” and published. These reports tell us which topics were discussed in what way and which members participated. The different activities which occur during such a meeting, in which we can analyze the politicians' behaviour, are [5]:

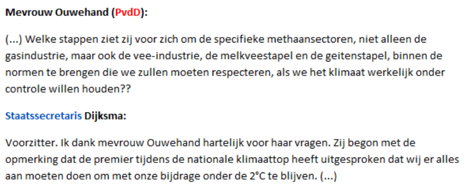
- Questions round (Vragenuur)
- Adjusting the current agenda (Regeling van Werkzaamheden)
- Plenary Debate
- Voting

Politicians can submit questions to the chairman of the plenary meetings. These questions can be addressed to any member of the government, meaning they are always addressed to the ministers. It is one of the primary ways used to control our government[6]. Analyzing these questions can give us an insight in how questions of politicians mutate during different events (public statements, passing of laws). When the chairman suggests adding items to the agenda, we can analyze the way politicians agree or disagree with the adjustment. The richest data contributing to our analysis is that of the plenary debates as it consists of the most speech data. After a debate there is a voting to make the House's final decision. Do the politicians' decisions reflect their actions?

Debates also happen in smaller groups: committees. Committees are formed by House members who are acquainted with the same subjects. Among the over 30 committees you have the ”Committee for Education, Culture and Science” and ”Committee for Finance”[7]. Analyzing the given reports grants an insight into a politician's behaviour when working in his domain, in contrast to the more general setting of the plenary meetings.

To know which topics could trigger a politician in any way, we will use his party's agenda and visions starting at the time of entering the House of Representatives. This data is (naturally) public as well.

Figure 4: Text sample of a questions round



**Mevrouw Ouwehand (PvdD):**

(...) Welke stappen ziet zij voor zich om de specifieke methaansectoren, niet alleen de gasindustrie, maar ook de vee-industrie, de melkveestapel en de geitenstapel, binnen de normen te brengen die we zullen moeten respecteren, als we het klimaat werkelijk onder controle willen houden??

**Staatssecretaris Dijksma:**

Voorzitter. Ik dank mevrouw Ouwehand hartelijk voor haar vragen. Zij begon met de opmerking dat de premier tijdens de nationale klimaattop heeft uitgesproken dat wij er alles aan moeten doen om met onze bijdrage onder de 2°C te blijven. (...)

[5] https://www.tweedekamer.nl/debat_en_vergadering/plenaire_vergaderingen

[6] https://nl.wikipedia.org/wiki/Kamervraag

[7] https://www.tweedekamer.nl/kamerleden_en_commissies/commissies

# 6 Requirements analysis

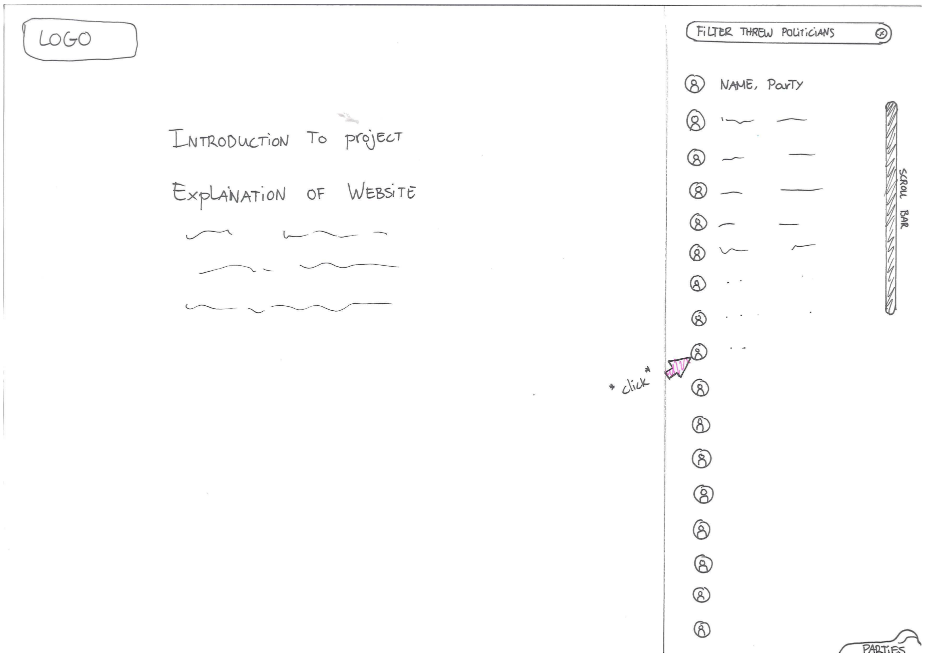
## 6.1 Project goal

With our product we want to grant a different insight into the House of Representatives to voters. In our web-application we want to create a dynamic but predefined overview that already gives some details. However, the user is free to select and examine different politicians and topics or alter filters in order to interactively find overviews that are interesting to him/her. By clicking around and filtering, the user will explore expected and unexpected relatedness and unrelatedness between topics, text features, political parties and politicians.

## 6.2 Description of Product

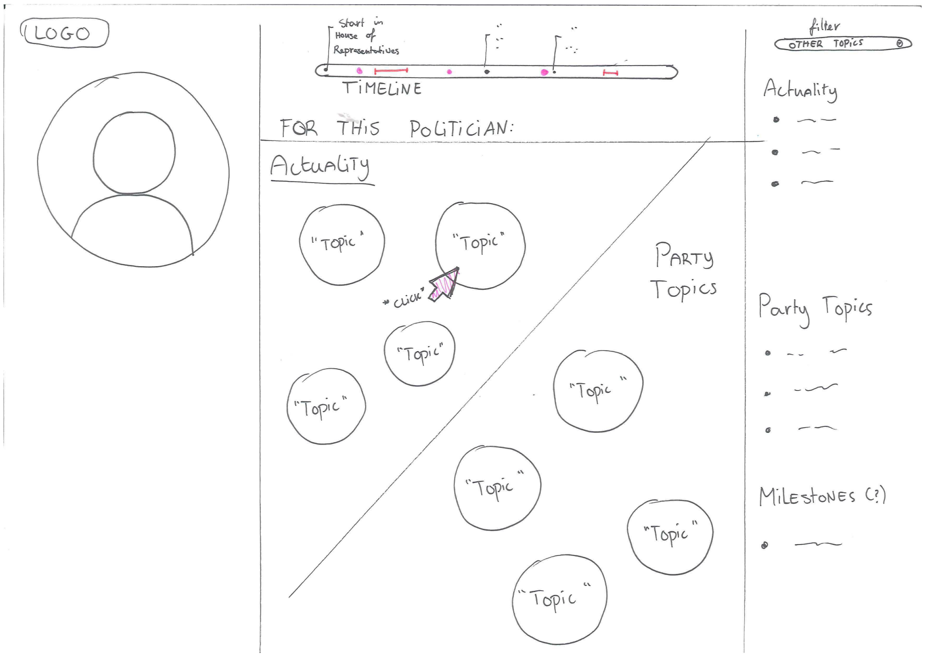
In figure 5 we can see the the so-called landing page. Our website can be seen as a dashboard for exploring the House of Representatives. The mid section is initially filled with brief information about our project and how to use this website. On the right side the menu for selecting a politician will already be open. When selecting a politician, the general information will be removed and the mid section will change accordingly.

Figure 5: Landing page



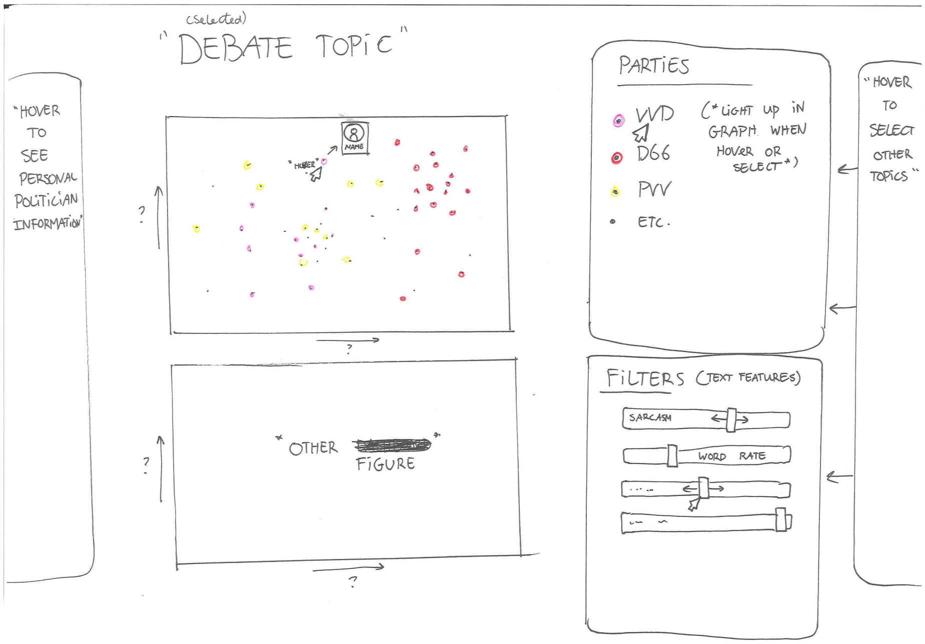
In figure 6 the page is shown after a user has clicked on the politician he or she wants to know more details about. The mid section now shows the topics (the circle-shaped labels in the midst of the screen) that are of great relevance to this particular politician. In the column to the left, we show a portrait of the politician, and underneath it, we include some personal information, for example the baseline, and some more insight of the party he or she belongs to. In the right column, we show a list of actuality (important recent incidents), his or her corresponding party topics, and important dates (milestones) to this particular politician.

Figure 6: When clicked on politician: topic selection



Once a topic has been chosen, the user will now see the page as in figure 7 the mid section will show the politician's position relative to his house members. The user can highlight the different politicians (dots) according to party or can filter them on different ratings for the text features.

Figure 7: When clicked on debate topic: dashboards


(selected)
"DEBATE TOPIC"
"HOVER TO SEE PERSONAL POLITICIAN INFORMATION"
"HOVER"
NAME
?
?
?
?
"OTHER FIGURE"
PARTIES
VVD
D66
PVV
ETC.
("LIGHT UP IN GRAPH WHEN HOVER OR SELECT")
FILTERS (TEXT FEATURES)
SARCASM
WORD RATE
"HOVER TO SELECT OTHER TOPICS"

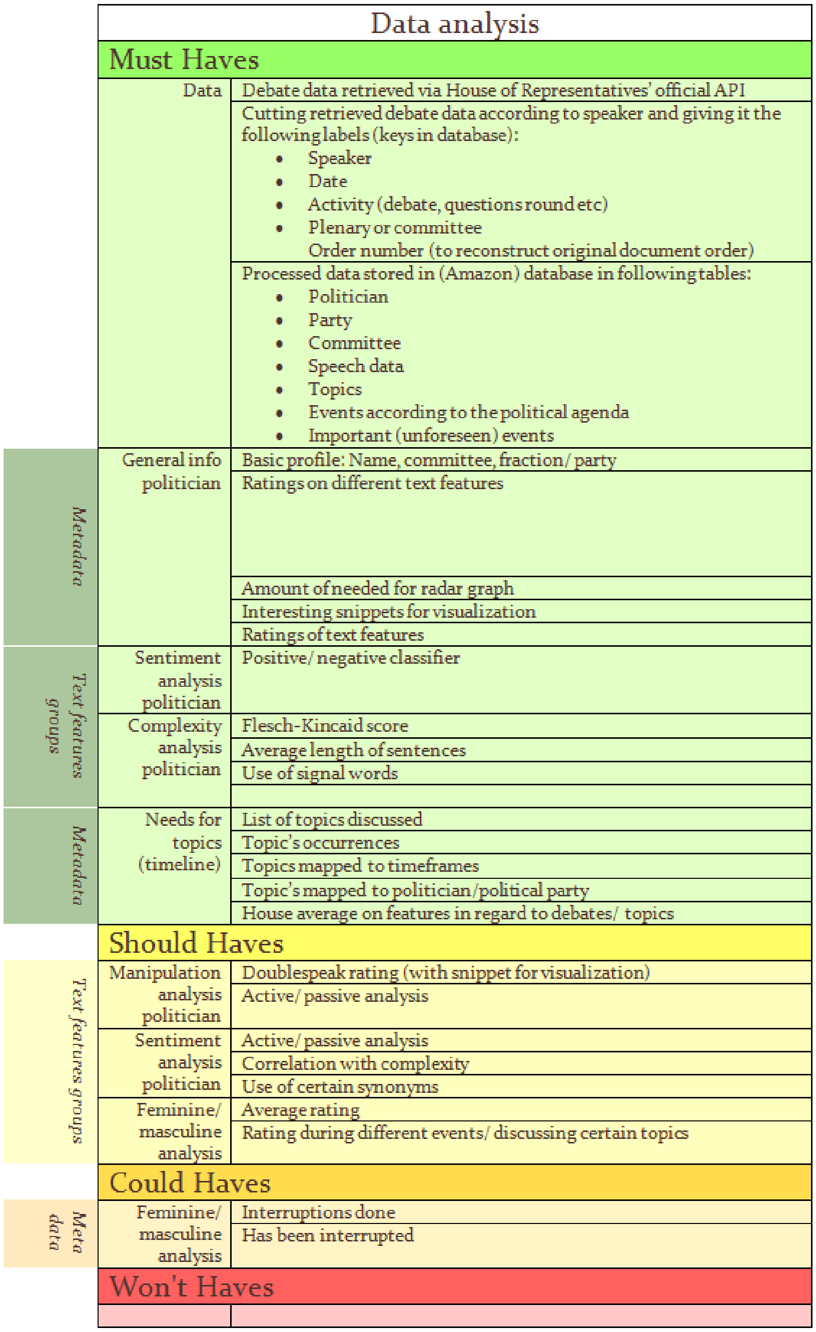
## 6.3 MoSCoW analysis

In order to manage our priorities we created MoSCoW tables. In Tables 8 and 9 you will find our requirements, prioritized using the MoSCoW method, for our data analysis and web-application respectively.

## 6.4 Scope and limitations

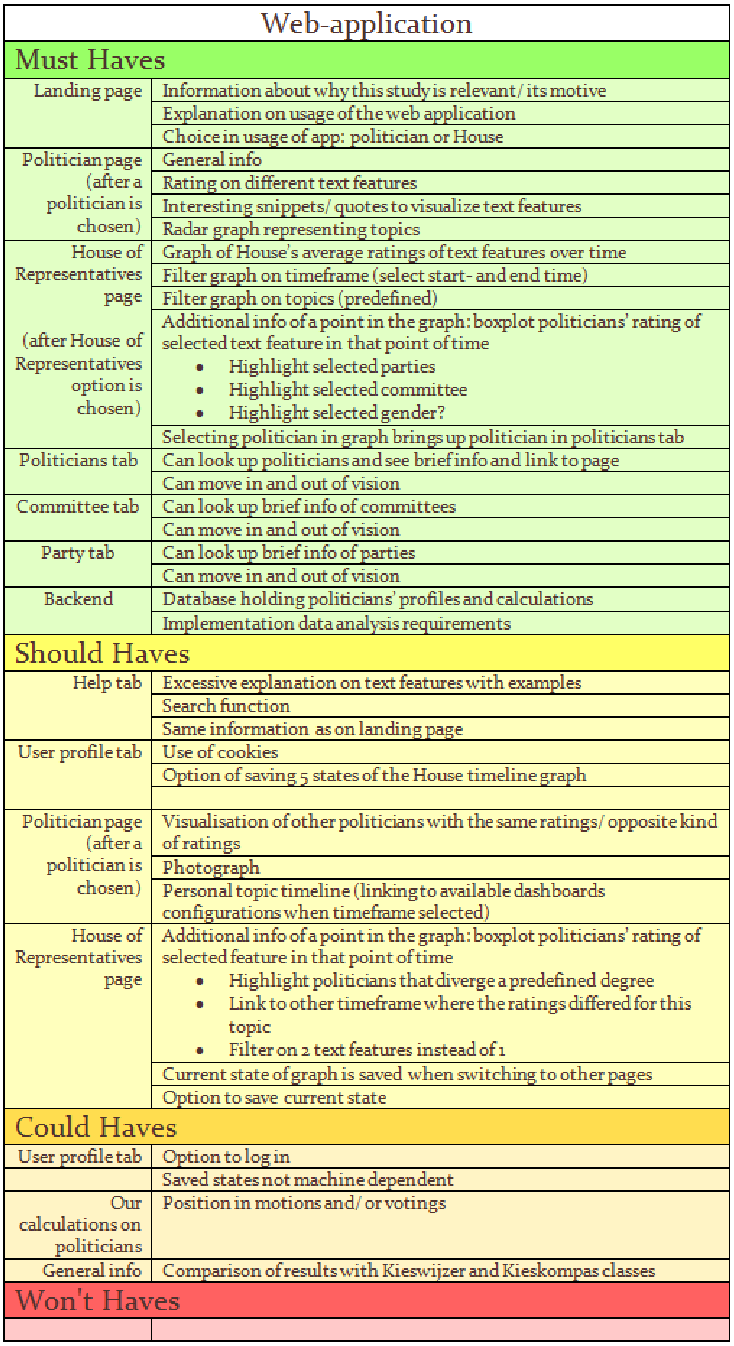
The product of our BEP will be a web page. We have a total amount of 9 weeks to complete the project. Not only do the literal words of a politician influence our opinion on the person, but also other aspects like for example; body language, tone of speech and facial expressions. Since we are provided with a limited amount of time, we will only focus on text.

Figure 8: MoSCoW prioritization table for Data analysis



| | Data analysis | |
|---|---|---|
| | **Must Haves** | |
| | Data | Debate data retrieved via House of Representatives' official API |
| | | Cutting retrieved debate data according to speaker and giving it the following labels (keys in database):<br>• Speaker<br>• Date<br>• Activity (debate, questions round etc)<br>• Plenary or committee<br>Order number (to reconstruct original document order) |
| | | Processed data stored in (Amazon) database in following tables:<br>• Politician<br>• Party<br>• Committee<br>• Speech data<br>• Topics<br>• Events according to the political agenda<br>• Important (unforeseen) events |
| *Metadata* | General info politician | Basic profile: Name, committee, fraction/ party |
| | | Ratings on different text features |
| | | Amount of needed for radar graph |
| | | Interesting snippets for visualization |
| | | Ratings of text features |
| *Text features groups* | Sentiment analysis politician | Positive/ negative classifier |
| | Complexity analysis politician | Flesch-Kincaid score |
| | | Average length of sentences |
| | | Use of signal words |
| | | |
| *Metadata* | Needs for topics (timeline) | List of topics discussed |
| | | Topic's occurrences |
| | | Topics mapped to timeframes |
| | | Topic's mapped to politician/political party |
| | | House average on features in regard to debates/ topics |
| | **Should Haves** | |
| *Text features groups* | Manipulation analysis politician | Doublespeak rating (with snippet for visualization) |
| | | Active/ passive analysis |
| | Sentiment analysis politician | Active/ passive analysis |
| | | Correlation with complexity |
| | | Use of certain synonyms |
| | Feminine/ masculine analysis | Average rating |
| | | Rating during different events/ discussing certain topics |
| | **Could Haves** | |
| *Meta data* | Feminine/ masculine analysis | Interruptions done |
| | | Has been interrupted |
| | **Won't Haves** | |
| | | |

Figure 9: MoSCoW prioritization table for final product



| Web-application | |
|---|---|
| **Must Haves** | |
| Landing page | Information about why this study is relevant/ its motive |
| | Explanation on usage of the web application |
| | Choice in usage of app: politician or House |
| Politician page (after a politician is chosen) | General info |
| | Rating on different text features |
| | Interesting snippets/ quotes to visualize text features |
| | Radar graph representing topics |
| House of Representatives page<br>(after House of Representatives option is chosen) | Graph of House's average ratings of text features over time |
| | Filter graph on timeframe (select start- and end time) |
| | Filter graph on topics (predefined) |
| | Additional info of a point in the graph: boxplot politicians' rating of selected text feature in that point of time<br>• Highlight selected parties<br>• Highlight selected committee<br>• Highlight selected gender? |
| | Selecting politician in graph brings up politician in politicians tab |
| Politicians tab | Can look up politicians and see brief info and link to page |
| | Can move in and out of vision |
| Committee tab | Can look up brief info of committees |
| | Can move in and out of vision |
| Party tab | Can look up brief info of parties |
| | Can move in and out of vision |
| Backend | Database holding politicians' profiles and calculations |
| | Implementation data analysis requirements |
| **Should Haves** | |
| Help tab | Excessive explanation on text features with examples |
| | Search function |
| | Same information as on landing page |
| User profile tab | Use of cookies |
| | Option of saving 5 states of the House timeline graph |
| | |
| Politician page (after a politician is chosen) | Visualisation of other politicians with the same ratings/ opposite kind of ratings |
| | Photograph |
| | Personal topic timeline (linking to available dashboards configurations when timeframe selected) |
| House of Representatives page | Additional info of a point in the graph: boxplot politicians' rating of selected feature in that point of time<br>• Highlight politicians that diverge a predefined degree<br>• Link to other timeframe where the ratings differed for this topic<br>• Filter on 2 text features instead of 1 |
| | Current state of graph is saved when switching to other pages |
| | Option to save current state |
| **Could Haves** | |
| User profile tab | Option to log in |
| | Saved states not machine dependent |
| Our calculations on politicians | Position in motions and/ or votings |
| General info | Comparison of results with Kieswijzer and Kieskompas classes |
| **Won't Haves** | |
| | |

# 7 Development Methodology

In this section we explain our approach on, and the structure of, our final system. Our decisions regarding data retrieval, processing and visualisation can be found here.

## 7.1 Retrieval of Data

Documented House meetings that are retrievable online date from 1995 and onwards. To limit our data regarding topics and different politicians, we will start our analysis with a recent House with completed data: 2012 - 2017. Fortunately, the House of Representatives released an updated API to access their publications. This will be our main way of retrieving data on political speech.
For the parties' (political) agendas we will be consulting their official websites, as we assume they publish official documents. All data will be stored in a database (see figure 10).

## 7.2 Toolkits

As stated earlier, we want to make use of sentiment analysis in order to profile and characterize our politicians. For this type of analysis, there exist so-called toolkits. A toolkit[8] is a collection of libraries that provide the user with a greater variety of possible solutions, as compared to the options the developer's environment initially supports. So, in order to avoid starting from scratch and reinventing the wheel, we sought a toolkit that enabled us to achieve an implementation that meets our requirements. In the following section, we provide a description about current toolkits that could contribute to such an implementation, discuss their pros and cons, and in the end justify our eventual choice. We did however also investigate toolkits that do not provide a sentiment analyser, but do actually offer components that could enhance the overall performance; a better sentence analyzer compared to our eventual choice.

### 7.2.1 Current toolkits

***Frog***[9] is powerful tool when it comes to sentence segmentation [30]. It is able to determine what grammatical type of words the sentence contains, providing a speech tag for each word the algorithm comes across. Here, it shows how 'certain' the assignment is by adding a confidence number. Frog thus also introduces the option to tokenize and lemmatize its input text. To us, Frog distinguished itself by the fact that it is completely based on the Dutch language. It uses corpora that contain hundreds of millions of words, and thus ensures any text written in Dutch can be thoroughly analyzed. The downside however, is that Frog is a finished product: extending it with a sentiment analyzer might

[8] http://stackoverflow.com/questions/5453011/api-vs-toolkit-vs-framework-vs-library

[9] https://languagemachines.github.io/frog/

prove to be extremely difficult. This is the main reason we decided not to initially use this toolkit.

***Natural Language Toolkit(NLTK)***[10] is a python based language processor composed of many libraries and programs that provide tools for sentence and sentiment analysis [31]. NLTK also allows easy access to more than fifty corpora and other databases focused on the classification of different languages. Furthermore, there is a lot of documentation on how the NLTK packages can be properly used, and how these packages can be combined in order to build a complete sentiment analyser. Regarding our requirements, the downside that NLTK has is that it does not initially support the Dutch language through its largest lexicon; WordNet, which revolves around the English language. A Dutch version[11] [32] of WordNet however does exist, so it should be possible to integrate this in the toolkit.

***Stanford NLP toolkit***[12] is also a very powerful tool that covers a lot of our requirements; speech tagging, named entity recognizing etc. [33]. Also, the tool is written in Java, which is the programming language we are most familiar with. The downside is that this toolkit does not seem to support anything written in Dutch. (We came across a sub-corpora download that enabled speech tagging in Dutch, but this library only contained around 150,000 words - as compared to 30 million for US English, so this seemed insufficient.)

### 7.2.2 Choice Justification

After considering the possible options each toolkit could provide, we decided to use NLTK. As stated, the NLTK comes with a lot of methods that could help realizing a desired implementation, also regarding both accuracy and efficiency. Furthermore, the existing documents clearly describe the necessary steps that are needed to be taken in order to achieve a basic sentiment analyzer. Since we are still in the very first phase of the project, it is of course possible that we will come across tools that might perform certain tasks better. Because of this, other toolkits, either mentioned here or new ones, could be used as well in the (near) future.

## 7.3 Knowledge bases

In the previous section we discussed some toolkits that could help achieve a sufficient or improved implementation, and in the end concluded that NLTK seemed the best option to start off with. As mentioned, this toolkit does not initially provide a very large database regarding the semantics of the Dutch language, so an additional download might be necessary. In this section, we will discuss such external semantic databases, and their potential contribution.

[10] http://www.nltk.org/

[11] http://wordpress.let.vupr.nl/odwn/

[12] https://stanfordnlp.github.io/CoreNLP/

### 7.3.1 Yago

Yago[13] or Yet Another Great Ontology, is a semantic database that combines the knowledge of other sources, such as Wikipedia, Geonames[14] and WordNet into one massive suite. This database holds a 95% confirmed accuracy, as its suite has been manually tested. Furthermore, Yago also maintains the structure of WordNet, which, as stated in the previous section, is already integrated in NLTK. Therefore, it should be possible to easily swap the initial version of WordNet for this expanded and improved database. In our case, the greatest property of this knowledge base however is that it supports the Dutch language, as it extracts Dutch Wikipedia pages in order to construct its suite. If we would thus be able to use Yago instead of NLTK's version of WordNet, this would make for a huge improvement. An extra aspect of Yago we might use is its option to cluster words based on thematic domains. That is, if it is supports themes related to the Dutch parliament.

### 7.3.2 DBpedia

Another knowledge base we looked at is a community powered project called DBpedia[15] (where DB stands for database). DBpedia aims to extract structures from Wikipedia pages and stores them into its own database. Users can then define their own specific query's and use these to retrieve desired information from the DBpedia suite. As stated in [34] this knowledge base can also be used for applications in the domains of Named Entity Recognition and topic detection. Like Yago, DBpedia supports the Dutch language because of its information retrieval through Wikipedia.

[34] also includes a section comparing Yago and DBpedia. Here, it is mentioned that Yago achieves higher precision and more consistent knowledge because it focuses on extracting a smaller number of relations, as compared to DBpedia. Also, since Yago makes use of WordNet rather than being manually powered, it contains more classes than DBpedia. For named entity databases, we thus preferred Yago.

### 7.3.3 SentiStrength

SentiStrength[16] is an application that allows the user to perform sentiment analysis over written texts. The algorithm assigns two values to each word in the text, the first varying between -1 and -5 to indicate negativity, and the second varying between 1 and 5 to indicate positivity. As a result, the text, or parts thereof, can be labeled with an overall mean 'score'. Although given the option on its website, SentiStrength was not initially able to analyse an example sentence written in Dutch, as none of the words were assigned a value. We added

[13] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[14] http://www.geonames.org

[15] http://wiki.dbpedia.org/about

[16] http://sentistrength.wlv.ac.uk/

an external list of Dutch words with matching sentiment score from an open source project on GitHub[17] and started the analyser again. This however still resulted in an obvious lack of basic sentiment words such as ’boos’ (meaning angry in Dutch), so we deemed this analyser to be insufficient. Furthermore, it is also explicitly stated on its website that SentiStrength could replicate human accuracy when it comes to short social texts, written in English, *except* political texts. This was the main reason we decided not to use this application.

### 7.3.4 Pattern

Another module that contains a sentiment database we looked at is called Pattern[18]. Pattern has tools for both data mining and natural language processing, including a sentiment analyser. Like SentiStrength, it returns a pair of values to assign an overall score to a piece of text. In Pattern however, the first value varies between -1.0 and 1.0 indicating polarity, while the second value, varying between 0.0 and 1.0, indicates subjectivity. The option to also get an estimate of a text’s subjectivity was one we deemed to be very interesting. Finally, Pattern supports Dutch. Overall, this module seems to meet our requirements, so as for sentiment databases, we preferred Pattern.

## 7.4 System Architecture

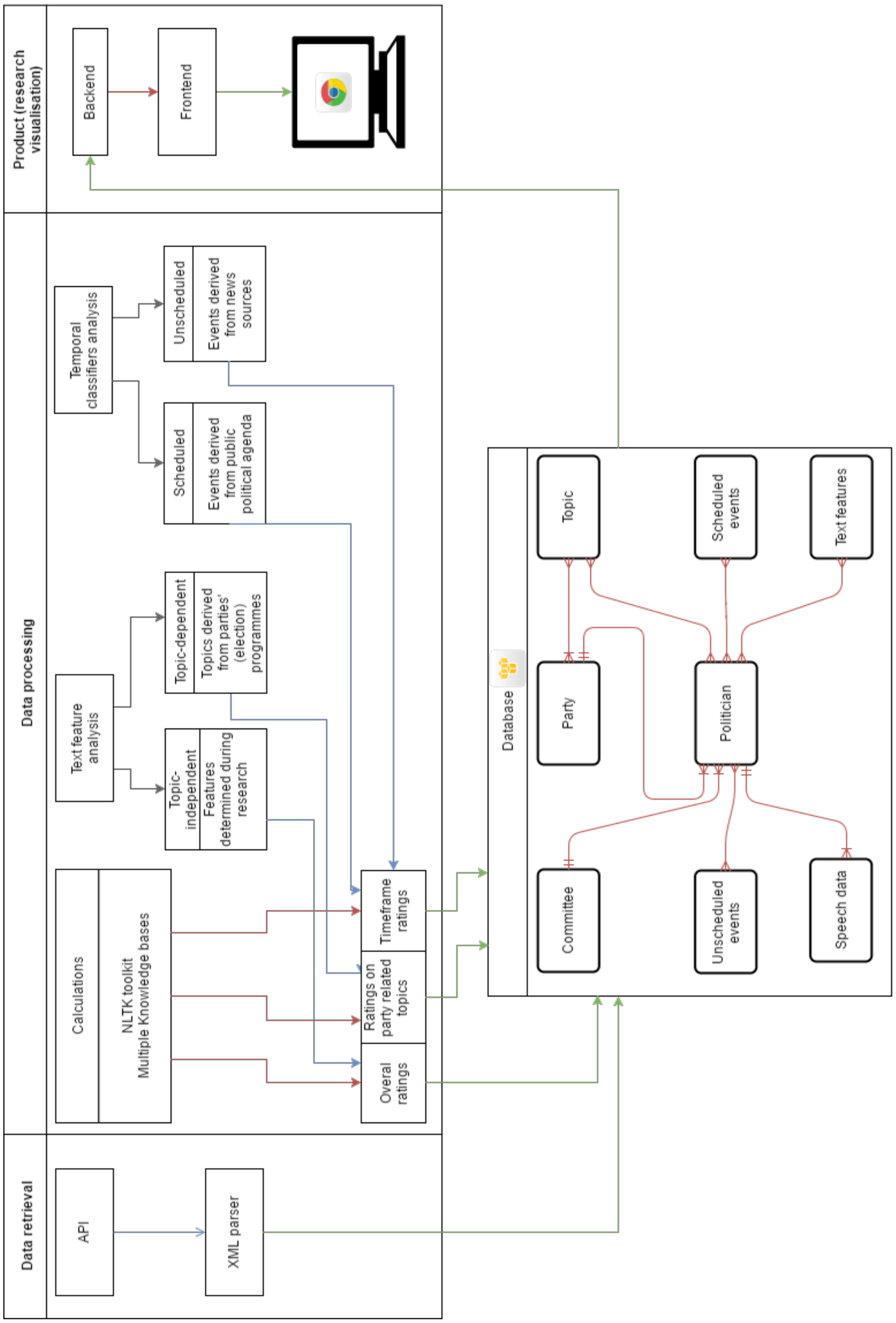
In figure 10 you will find an overview of our system’s architecture using Crow’s Foot Notation [19].

[17] https://github.com/felipebravom/StaticTwitterSent/tree/master/extra/SentiStrength/dutch

[18] http://www.clips.ua.ac.be/pages/pattern-nl

[19] https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

Figure 10: System's Architecture



# 8 Success Criteria

The requirements for our research and consequential product are stated in Section 6, mainly in tables 8 and 9. In means of deliverables we would like to implement at least the requirements stated in the ”Should have” sections of the tables. Our process, and therefore our implementations, should be well documented in a report which will be delivered as well. In some way the report itself is the most important as, at the beginning of our project, we do not know what (meaningful) results will be discovered during different phases. Or, if they will be found at all. Thus we should be able to provide support as to why and how this outcome could occur. In general, our findings, whether eventual or partial, should be accompanied by explanation through analysis, and, if possible, earlier results.

## 8.1 Analysis with text features

### 8.1.1 Text features

After relevant text features have been chosen, their contribution to our research (questions) should be shown. Does this text feature, on its own or in combination with other text features, enhance our results? Does this text feature actually give more insight into a politician’s speech pattern? If these questions can be answered positively, success is met. If this is not the case, we would want to document it as well to show why it was not included.

### 8.1.2 Topics and temporal influences

We need to find topics and different points in time which have an impact on a politician’s speech pattern; cause a variance. We expect that, for each party in particular, topics which are of great(er) importance are likely to cause such a variance. For temporal influences we are not bounded by the party, as important periods or events apply to the whole house. Here, success criteria will be met if we succeed in finding cases that induce variances. However, if (expected) variances are not introduced, we will provide an explanation. If this is deemed to be impossible, a reasoning as to why this is the case will suffice.

## 8.2 Visualisation

When it comes to visualization, it is crucial that we present our findings in a clear way to the user. The latter should, through quite straight-forward steps, be able to request information without any confusion being introduced. Here, we must focus on structuring the website in such a way that it provides easy access to its tools and data. Naturally, it is also very important the user understands the visualized results of his or her query. Therefore, we have to determine which visualization method(s); graphs, diagrams, box-plots, etc., best suit the type of information to be presented. The combination of these two aspects describes the success criteria for this particular component; namely, the application should be easy to use, and the overall goal we want to achieve - making our research accessible and therefore giving a new form of insight in the House of Representatives - should become clear.

# 9 Performance Testing

Testing the quality of our software is an obvious necessity throughout the entire development process. During each phase, we should test its stability and determine whether its generated output is correct or not. Since it is not exactly known what our eventual results will look like however, numerous testing methods cannot be fully applied to our particular application.

For example, Black-box testing would require a set of correct results, which could then be used to validate the output of our application. Although we will be able to point out extreme cases manually, this output will always vary in a certain range, and in our case, the output will most definitely not be the same when looking at multiple politicians. When considering the Flesch-Kincaid score for example, we expect it to be between 6 and 12 for any politician. We deem these to be valid results. If any outliers are detected, we can trace back their origin and determine if this is due to an error in our software or not. The Black-box testing sub-technique that is most similar to this approach is focussing on boundary values; define a range of values (a minimum and maximum), and verify if our results can be mapped into it.

White-box testing however can be applied in many instances. Its testing techniques, control-flow and data-flow testing, allow us to improve the structure and efficiency for each phase during the development process. In order to perform control-flow testing, we will define test cases that cover as many - preferably all - of the statements within each component of the application; a set of paths. These paths could for example help us detect unreachable code. For data-flow testing, we could define test cases that cover all variables per component of our application, and determine if they were correctly changed, initialized, etc. Again, as stated earlier, it is not possible to fully ensure correctness of the generated output, which in turn might leave some smaller errors in our code to be unnoticed. This also impedes the integration of Unit Tests; even on component level, correct results may be impossible to be perfectly defined.

As for Performance testing, we can insert a very great amount of data - an amount that could however be a possible situation in extreme cases - and check if the algorithm does not get overloaded or that information will be lost during the analysis. In our case, we want to get as much input as we can; all documented speeches during debates per politician - so this testing method is of great importance.

We will also focus on Regression testing; we can run an updated or changed component again and verify if its output didn't change compared to its previous state. Here, we can determine whether the mutation had a negative (or positive) effect on the application.

Finally, we will test the users' experience of the application by means of their interaction with the interface provided by the corresponding website. In this Acceptance testing phase, we will first determine for ourselves whether we deem the layout, functionality, readability etc., to be sufficient or not, and in the following stage, let a group of users test our application and process their feedback.

In general, these different types of testing ensure that the application will be thoroughly tested and maintained throughout the entire development process, and that its final version will meet as many of our proposed requirements.

# References

[1] A. Hunt and B. Wheeler, "Brexit: All you need to know about the uk leaving the eu," April 2017. [Online]. Available: http://www.bbc.com/news/uk-politics-32810887

[2] "Brexit: Europe stunned by uk leave vote," June 2016. [Online]. Available: http://www.bbc.com/news/uk-politics-eu-referendum-36616018

[3] L. Wentz, "No brexit-like surprise victory is ahead for trump, says pundit who got brexit right." [Online]. Available: http://adage.com/article/campaign-trail/trump-pull-a-brexit-surprise-victory/306385/

[4] "Who said brexit was a surprise?" June 2016. [Online]. Available: http://www.economist.com/blogs/graphicdetail/2016/06/polls-versus-prediction-markets

[5] "Official 2016 presidential general election results," January 2017. [Online]. Available: http://www.fec.gov/pubrec/fe2016/2016presgeresults.pdf

[6] P. Healy and J. W. Peters, "Donald trump's victory is met with shock across a wide political divide," November 2016. [Online]. Available: https://www.nytimes.com/2016/11/10/us/politics/donald-trump-election-reaction.html?_r=0

[7] N. Jackson and A. Hooper, "Forecast president senate," November 2016. [Online]. Available: http://elections.huffingtonpost.com/2016/forecast/president

[8] "General election trump vs clinton. (n.d.)." [Online]. Available: http://www.realclearpolitics.com/epolls/2016/president/us/general_election_trump_vs_clinton-5491.html

[9] E. Bell, "The truth about brexit didn't stand a chance in the online bubble," July 2016. [Online]. Available: https://www.theguardian.com/media/2016/jul/03/facebook-bubble-brexit-filter

[10] E. Bakshy, S. Messing, and L. Adamic, "Exposure to ideologically diverse news and opinion on facebook," *Science*, vol. 348, pp. 1130–1132, 2015.

[11] R. D. Young, "The google "filter bubble" and its problems," May 2011. [Online]. Available: https://www.searchenginejournal.com/the-google-filter-bubble-and-its-problems/29879/

[12] "Survey: News use across social media platforms 2016," January-February 2016. [Online]. Available: http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/

[13] C. Doherty, “7 things to know about polarization in america,” June 2014. [Online]. Available: http://www.pewresearch.org/fact-tank/2014/06/12/7-things-to-know-about-polarization-in-america/

[14] L. Bode, E. K. Vraga, P. Borah, and D. V. Shah, “A new space for political behavior: Political social networking and its democratic consequences,” *Journal of Computer-Mediated Communication*, vol. 19, pp. 414—429, 2013.

[15] M. R. Mehl and K. G. Niederhoffer, “Psychological aspects of natural language. use: our words, our selves.” *Annual Review of Psychology*, vol. 54, pp. 547–577, 2003.

[16] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, “Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel,” DTIC Document, Tech. Rep., 1975.

[17] R. Prabowo and M. Thelwall, “Sentiment analysis: A combined approach,” p. 1, 2009.

[18] G. Katz, N. Ofek, and B. Shapira, “Consent: Context-based sentiment analysis,” *Knowledge Based Systems*, vol. 84, pp. 162–178, 2015.

[19] A. Hassan, H. Korashy, and W. Medhat, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Engineering Journal*, vol. 5, pp. 1093–1113, 2014.

[20] W. E. Lutz, *Beyond Nineteen Eighty-Four: Doublespeak in a Post-Orwellian Age.* Princeton University Press, 1989.

[21] C. F. Karpowitz and T. Mendelberg, *The Silent Sex Gender: Deliberation, and Institutions*. Princeton University Press, 2014.

[22] J. J. Jones, “Talk “like a man”: The linguistic styles of hillary clinton, 1992–2013,” *Perspectives on Politics*, vol. 14, no. 3, pp. 625–642, 2016.

[23] “Gender inequality index,” 2015. [Online]. Available: http://hdr.undp.org/en/composite/GII

[24] “Women in national parliaments,” March 2017. [Online]. Available: http://www.ipu.org/wmn-e/classif.htm

[25] B. Yu, S. Kaufmann, and D. Diermeier, “Classifying party affiliation from political speech,” *Journal of Information Technology & Politics*, vol. 5, no. 1, pp. 33–48, 2008.

[26] G. Bohner, “Writing about rape: Use of the passive voice and other distancing text features as an expression of perceived responsibility of the victim,” *British Journal of Social Psychology*, vol. 40, no. 4, pp. 515–529, 2001.

[27] N. M. Henley, M. Miller, and J. A. Beazley, “Syntax, semantics, and sexual violence agency and the passive voice,” *Journal of Language and Social Psychology*, vol. 14, no. 1-2, pp. 60–84, 1995.

[28] E. Tarone, S. Dwyer, S. Gillette, and V. Icke, “On the use of the passive and active voice in astrophysics journal papers: With extensions to other languages and other fields,” *English for specific purposes*, vol. 17, no. 1, pp. 113–132, 1998.

[29] D. R. Olson and N. Filby, “On the comprehension of active and passive sentences,” *Cognitive Psychology*, vol. 3, no. 3, pp. 361–381, 1972.

[30] A. V. Bosch, M. V. Gompel, I. Hendrickx, and K. V. Sloot, *Frog: A Natural Language Processing Suite for Dutch*, 2016.

[31] S. Misailovic and K. Yessenov, “Sentiment analysis of movie review comments,” pp. 1—9, 2009.

[32] E. van Miltenburg, M. Postma, R. Segers, A. Schoen, and P. Vossen, “Open dutch wordnet,” pp. 300–308, 2016.

[33] J. Bauer, S. Bethard, J. Finkel, C. Manning, D. McClosky, and M. Surdeanu, “The stanford corenlp natural language processing toolkit,” 2009.

[34] S. Auer, C. Bizer, S. Hellmann, R. Isele, J. Robert, M. Jakob, A. Jentzsch, P. van Kleef, D. Kontokostas, J. Lehmann, P. Mendes, and M. Morsey, “Dbpedia - a large-scale, multilangual knowledge base extracted from wikipedia,” 2012.

[35] M. Toplak and A. N. Katz, “On the uses of sarcastic irony,” *Journal of pragmatics*, vol. 32, no. 10, pp. 1467–1488, 2000.

[36] B. Keysar, “The illusory transparency of intention: Linguistic perspective taking in text,” *Cognitive psychology*, vol. 26, no. 2, pp. 165–208, 1994.

[37] V. Hatzivassiloglou and H. Yu, “Toward answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences,” *EMNLP ’03 Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129–136, 2003.

[38] V. Hatzivassiloglou and Y. M. Wiebe, “Exposure to ideologically diverse news and opinion on facebook,” *COLING ’00 Proceedings of the 18th conference on Computational linguistics*, vol. 1, pp. 299–305, 2000.

# A Text Features

In this section we present text features that were considered, but not implemented.

### Sarcasm

Sarcasm indicates indirect criticism towards the addressee. In [35] it is shown that sarcasm is, apart from being used as a humorous tool, often used as a means of verbal aggression (mockery, anger-provoking, impoliteness). Sarcasm can also enhance memorability. Detection of sarcasm depends on perspective; knowledge of the topic discussed and the scenario in which it's discussed [36]. Since we will be analyzing text, we wont have intonation or body language indicating sarcasm. We need to take the perspective of the speaker and addressee into account to determine the presence of sarcasm (in our case the whole house or members of the parliament). By using the background of the ones involved we can measure if sarcasm would be detected and/ or if it is used intentionally.

### Subjective/objective

In [37] it is stated that in order to train a Naive Bayes algorithm for sentiment detection, features are included that count the amount of positive and negative words in a sentence. This is because [38] argues that when such words are used in a sentence, it can be an indication that the particular sentence is subjective. This is because opinions are more likely to be expressed through words with a sentimental value rather than ones without.

### Speech rate

By analyzing the speech rate, we may find which topics in particular affect a politician.