JOHANNES KEPLER
UNIVERSITY LINZ | JKU

TNF

Faculty of Engineering
and Natural Sciences

Understanding the Community Structure of Coalition and Opposition in Parliamentary Democracy on the Example of the Austrian Parliament

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements
for the academic degree

Bachelor of Science

in the Bachelors Program

Computer Science

Submitted By:
Markus Hiesmair

At the:
Institut für Telekooperation

Advisor:
Gabriele Anderst-Kotsis

Linz, 20. November, 2015

Affidavit

I hereby declare that the following bachelor's thesis ”Understanding the Community Structure of Coalition and Opposition in Parliamentary Democracy on the Example of the Austrian Parliament” has been written only by the undersigned and without any assistance from third parties.

Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

Linz, on December 14, 2015

Markus Hiesmair

Acknowledgment

Summary

Summary . . .

Abstract

Abstract ...

Contents

Abbreviations

BZÖ Austrian Future Party (german: Bundeszukunft Österreichs)

FPÖ Austrian Freedom Party (german: Freiheitliche Partei Österreichs)

Grüne The Greens (german: Die Grünen)

Liberale Liberal Forum (german: Liberales Forum)

ÖVP Austrian People Party (german: Österreichische Volkspartei)

SPÖ Social Democratic Party of Austria (german: Sozialdemokratische Partei Österreichs)

List of Figures

List of Tables

Chapter 1

Introduction

One of the most crucial requirements of a democracy is transparency. There are several ways how one can gain information about the current and past political activities in Austria. One of the best possibilities among them are the publicly available protocols of the national council sessions. In these protocols every word said in a session is written down and that makes up the corresponding protocol. Unfortunately, these protocols are very long and it is hard to gain meaning out of it, because of its plain and simple structure and the great amount of data.

To be able to analyze and visualize the activities and relations of the politicians and parties in a better way, and to make the structure of the political system accessible to a broader audience, analysis tools are needed. This thesis documents the methods that can be used to perform automated analysis over the available data. The protocols are being extracted, transformed, analyzed and visualized.

1.1 Research Goals

The protocols are currently available in semi-structured form - through HTML files.[1] To be able to properly persist and analyze the data, the protocols have to be transformed into a fully structured form (e.g. Java Objects). The following elements will be extracted:

- Legislative periods and their sessions
- Politicians and their mandates

[1] Until the 19. legislative period, the protocols are only available in PDF-format. These protocols cannot be extracted with reasonable quality. Therefore they won't be used in the further work.

- Parliament clubs
- Discussions and speeches during the sessions

As soon as this is done, the extracted data can be persisted into an arbitrary relational database. Furthermore, some general and network analysis should be done on the data. In the following list some interesting applications on top of the extracted data are presented:

- Create a network graph which shows the relations among politicians and parliament clubs.
- Find groups of politicians (or parliament clubs) with the same attitudes.
- Analyze how homogeneous the attitudes of politicians of the same parliament club are.
- Find the politicians which take part in the most discussions.
- Find the most absent national council members.

In the final step the results should be visualized via a web application. The focus hereby lies in making the results as easy to understand as possible.

1.2 Austrian Parliament

The analysis approaches in this thesis can basically be used for every parliament or other similar political structure, given that data is available in a similar form. As this work is done at an Austrian university and the protocols of the national council are available as open data, the show case is built up on the Austrian parliament.

The Austrian parliament basically consists of two chambers, the national council and the federal council. The national council is elected through federal elections, whereas the federal council consists of delegates of the 9 provinces. Both chambers have different responsibilities and functions, and their goal is to ensure that the decisions are in the best interest for the Austrian people [8].

1.2.1 National Council

The national council consists of 183 members, which can band together to form so called parliamentary clubs. Usually for each political party, which got elected in the national council, there is one parliamentary club, but that is no necessity. The tasks of the national council include law-making, controlling the government, seeking solutions for current problems, determining the budget and much more [8].

After every election one or more parliamentary clubs have to build the government. These parliament clubs have to have in total at least 92 mandates (more than half of the overall mandates of 183). Politicians of the government are then selected to be the federal ministers. All other parliamentary clubs, which are not in the government are in the so called opposition. In most cases the government can make laws without the opposition, but in some special cases there is $\frac{2}{3}$ majority needed [8].

1.2.2 Federal Council

The federal council consists of 61 members. As the members are delegates of the provinces, their main duty is to represent their province and make sure the politics in the parliament are in the interest of the province they represent. To do so, they can raise objections against legislation of the national council, but most of the time the federal council only has the power to delay legislation and not to prevent it [8].

1.2.3 Analysis Scope

In this work only the data of the national council will be analyzed because there are no openly available data sources which could be used to include the federal council in the analysis. Furthermore, the national council has a lot more responsibilities and is of greater importance for the overall democratic process in Austria.

Chapter 2

Related Work

In the context of computer science, there is only little work on automatically analyzing political structures such as a parliament. In 2013, Renzo Lucioni [7] used publicly available voting data from the Congress of the United States of America to analyze the relationships among politicians and to determine how distinct the two main parties, the Democrats and the Republicans, are. To achieve this, he used data from the 101^{st} Congress through the 113^{th} Congress and created network graphs which graphically showed which politicians vote similar. He also showed how the structure of the Congress developed over time by creating several graphs of the years 1989 to 2013. His results showed that the gap between the Republicans and the Democrats became larger and larger over the last decades. This means that both parties vote more and more against each other. In the context of the Austrian parliament, similar analysis can be applied, if data is available in sufficient quality. For example, it can be analyzed which parties vote similar and if there exist relations between parties which are in coalition and parties which are in opposition. You can find the results for the Austrian parliament in section 4.1.

An earlier work was done by Porter and Newman in 2005 [9]. They did analysis of committees in the U.S. House of Representatives and tried to show the connections between representatives of the House and the committees and subcommittees. Most of the U.S. legislation is discussed and decided in these committees and especially the assignment of politicians and the change of it over time are interesting subjects of analysis. In their work Porter and Newman gain information without specific knowledge of the structure of the committees, using technologies of network analysis. In particular, they tried to find communities and their connections within the network of the committees to get information about strategic assignment of politicians in important committees. Furthermore, Porter and Newman used single-linkage clustering to get clusters of communities and their connections and also visualized that with a dendrogram[1] representing the hierarchical structure of the committees and subcommittees. Similar analysis would also be interesting for the Austrian parliament, but community detection and clustering are not included within the scope of this thesis. In the second part of Porter and Newman's paper, they also have done some analysis on the relations among politicians in the House of Representatives. The results show the most left, most right and most partisan politicians in the house. This is especially interesting because it shows that all most left politicians are Democrats and all most right politicians are Republicans, which shows that also the House of Representatives is divided at a high degree into Democrats and Republicans.

In 2012, Amelio [1] did a similar study on the voting behavior in the Italian parliament. One part of her study was analyzing party cohesion (how homogeneous all politicians of a specific party voted in the selected periods). An interesting result was that the cohesion of the parties in the opposition increased over time whereas the cohesion of the governing parties decreased and after the analyzed period the government was not reelected. Another measure taken was the parliamentarian similarity. This measure compares the voting behaviors of two parliamentarians and gives a result on how similar they voted. Based on the values obtained, Amelio did hierarchical clustering using single-linked clustering to find communities within the parliament and visualized the results in a dendrogram, similar to the result of Porter and Newman [9].

All three papers, which were discussed, show that through automatic analysis of political structures, information on the structure and clustering of political systems can be gained. Furthermore, through visualizations in graphs the information can be presented in a way understandable for laymen and therefore the visualizations can be used to improve the general understanding of political systems and the current structures of parties and politicians.

2.1 Related Work in the context of the Austrian Parliament

Austria is no country with good information laws in terms of governmental transparency. In the last five years (from 2011 to 2015) in the ”Right To Information”-Rating Austria was on the last place each time and in 2015 there were 102 nations investigated on their governmental transparency laws [5].

To improve the transparency of political and governmental processes in Austria, in 2010 an organization with the title ”Informationsfreiheit” (Information freedom) [5] was founded. They held several online campaigns and fought in several law suits for

[1] A dendogram is a tree diagram, which is often used as representation for results of hierarchical clustering.

the improvement of the right to information and already had quite good success. Furthermore, they started a project called ”OffenesParlament.at” (open parliament) [6] which has the goal to give a better overview of the data which is available on the Austrian parliament website [8]. Therefore, they also extract the required information out of the openly available HTML documents used in the context of this thesis. On their website, there will be a lot of documents of the parliamentary process available and it will presented and a more user friendly way. A few of the items presented are: discussions grouped by topic, speeches of politicians, bills and bill drafts and statements of politicians to bill drafts. As you can see the data, which OffenesParlament.at extracts is mostly the same which is extracted within this thesis.

Chapter 3

Design and Implementation

In this chapter the design and implementation of the prototype for the Austrian parliament are described. First of all, in Section 3.1 the overall architecture and the different components are being discussed. The more detailed description of the implementation is divided into five sections: Section 3.2 describes the data extraction from HTML-files, section 3.3 describes the transformation into a structured form, section 3.4 discusses the export to a relational database and the last sections describe the analysis and visualization of the given data.

3.1 Architecture

Figure 3.1 shows the general architecture of the prototype which was implemented. The Extract-Transform-Load-Application (ETL-Application) brings the data from the debate transcripts in the database whereas the web server application visualizes the results and shows statistics and graphs for the given data. The ETL-Application is implemented using the ETL pattern. This means that there are three distinct steps: Extract - Transform - Load. First the application reads an RSS feed which contains all the transcripts for one legislative period and the politician profiles (Extract). The retrieved HTML-files get parsed and are transformed into Java objects (Transform) which get loaded into a relational database[1] (Load). To visualize the then available data, the analysis engine queries the database, performs analysis on it and converts the data in a form which can be displayed (e.g. a graph structure). Furthermore, this data is made available via RESTful web services. The Polymer web application[2] accesses these web services and shows graphs and statistics. All the components will be described in more detailed in the following sections.

[1] in the prototype, a PostgreSQL database was used

[2] See section 3.6.2.

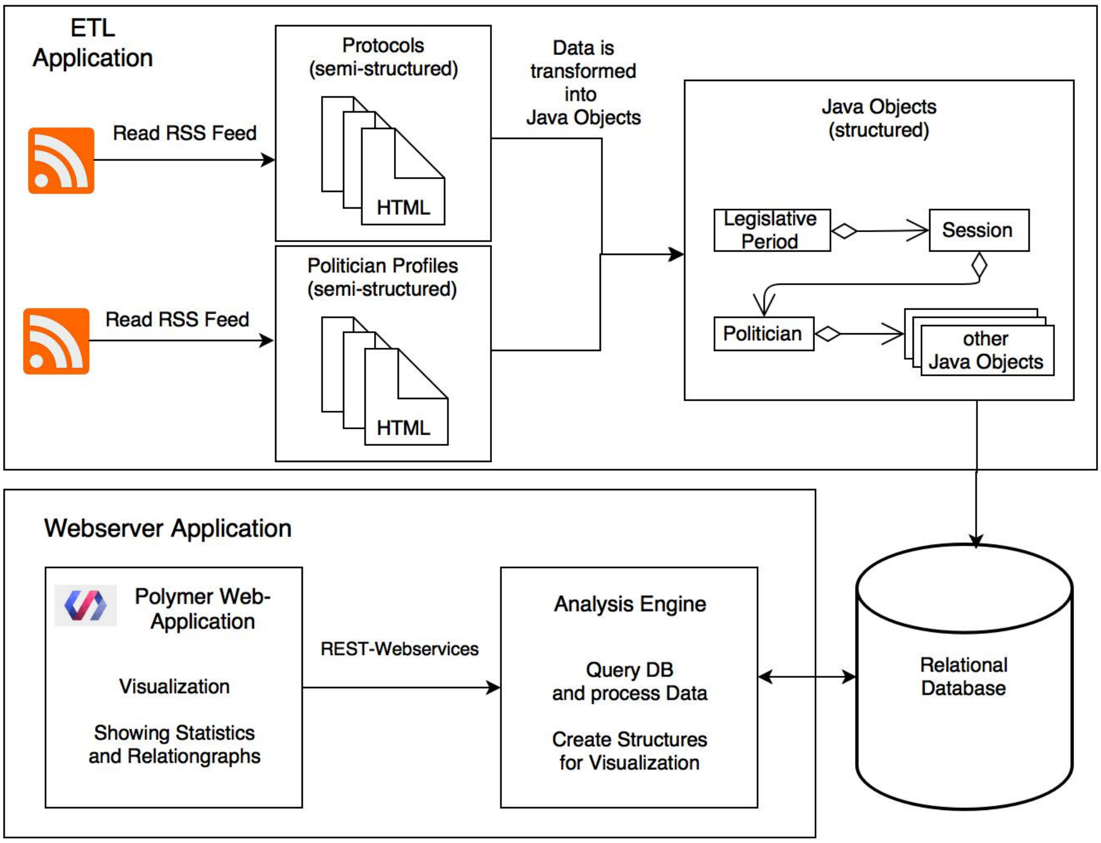

ETL Application
Read RSS Feed
Protocols (semi-structured)
HTML
Read RSS Feed
Politician Profiles (semi-structured)
HTML
Data is transformed into Java Objects
Java Objects (structured)
Legislative Period
Session
Politician
other Java Objects
Webserver Application
Polymer Web-Application
Visualization
Showing Statistics and Relationgraphs
REST-Webservices
Analysis Engine
Query DB and process Data
Create Structures for Visualization
Relational Database

Figure 3.1: General Architecture

3.2 Data Extraction

The first step which has to be done in the ETL-Application is the extraction. The data which is to be transformed has to be collected and stored. In our case the data is contained in the stenographic transcripts of the national council and in politician profiles. Both the transcripts and politician profiles are publicly available and can be found at the website of the Austrian parliament (See `https://www.parlament.gv.at/PAKT/STPROT/` and `https://www.parlament.gv.at/WWER/PARL/`). The transcripts are available in PDF-format and since the 20^{th} legislative period also in HTML. As the transforming of the PDF-files would not result in sufficient quality, in this thesis only the HTML-files (the data since the 20^{th} legislative period) are being extracted and analyzed.

To collect the needed files automatically, RSS feeds are used. There are feeds available for both, stenographic transcripts and politician profiles. The HTML-files which are linked in the feeds are being downloaded and stored on the file system before they are transformed.

3.3 Transformation

The next step is the transformation of the HTML-files into Java objects (transformation of semi-/unstructured data to structured data). To be able to extract the desired information out of the HTML-files, the structure of the stenographic transcripts and politician profiles was investigated. Then undesired HTML-structures were removed, because they avoided a correct information extraction. For example, the page breaks and page headers in the full text transcripts were removed. Finally, the tag structure of the HTML documents and regular expressions were used to find the required data. This has proven to be a very challenging task since the transcripts changed over time. For example, in the full text transcripts since the 21^{st} legislative period, the politicians were referenced with a link to their profile, whereas in the transcripts of the 20^{th} period, there were only politician names. This made it much harder to find the right politicians as the full list of politicians had to be searched for a politician with the same name and title(s). To transform the HTML-files with good quality, for the transcripts of the 20^{th} period, there was an extra transformation step which could find the politicians without the link to their profiles.

3.3.1 Transformation of the Politician Profiles

For each politician who is sitting in the national or federal council, there exists a politician profile. In the first part of the transformation, all politician profiles are being transformed into a list of politician Java objects. The name (and if provided previous names), the titles, the birth date, and the political mandates are being extracted from the HTML code. Mandates are some kind of political functions like the membership in the national or federal council or the period where the politician was a federal minister. They are of special importance because they include the club memberships of the politician within a specific period of time.

3.3.2 Transformation of the Transcripts

In the second part of the transformation, the transcripts of the sessions of the national council are transformed. For each session, there exist two files: The full text transcript and the session summary. In the full text transcript every word spoken in a debate is recorded, whereas in the session summary, there is only important information like discussions and speeches summarized. Information that gets extracted out of the transcripts includes sessions of legislative periods, chair men in the sessions, politician absences and presences, discussions and speeches of politicians. The list of politicians which was the result of the first transformation part is used to find the politicians in the transcripts and to reference them in speeches. Figure 3.2 shows the class diagram of the objects which were extracted in both transformation steps and their relations.

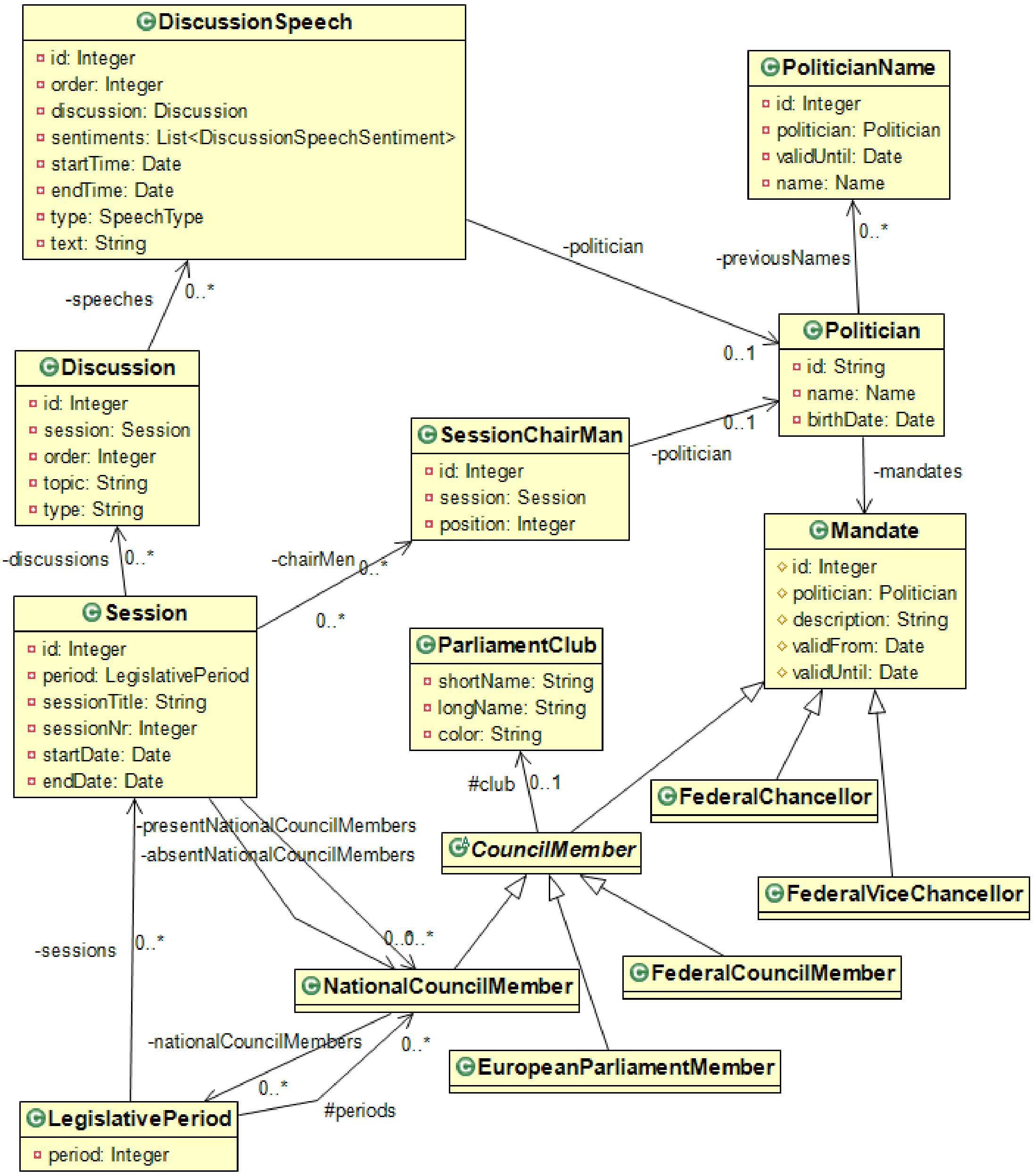


Figure 3.2: Class Diagram of the result of the Transformation Step

3.4 Export into a Database

The export to a database is the loading part of the ETL-application. The Java objects which were created in the transformation step are being persisted into a relational database. To stay independent of specific databases the Java Persistence API implementation Hibernate[3] is used. Using OR-mapping brings the advantages that no SQL-statements have to be written and changes on the tables/objects are easily made using Java annotations.

3.5 Analysis

The first step of the analysis phase was to determine which analysis could be applied on the given data. The results can be classified into two categories: Simple analysis and network analysis. The simple analysis takes just simple measures like how many speeches a politician held during a legislative period or how often was he absent. These measures are taken via SQL queries on the database built up by the ETL Application. Table 3.1 shows all simple analysis measures taken. The network analysis will be discussed in the following sections.

Name	Description
Overall Absence per Period	A percentage of the absence of all national council members in one legislative period.
Absence of a Politician per Period	A percentage of the absence of one national council member in one legislative period.
Absence of a Parliamentary Club per Period	A percentage of the absence of all members of one parliamentary club in one legislative period.
Count of Speeches per Politicians and Period	The count of speeches a politician held in the national council during one legislative period.
Mandate Distribution per Period	The assignment of the 183 political mandates to the parliamentary clubs.
Average Serving Duration in the National Council	How long does the average politician serve in the national council.
Longest Serving National Council Members	Which politicians serve the longest in the national council.

Table 3.1: Simple Analysis Measures

3.5.1 Politician Relation Graph

Network analysis similar to the one Lucioni [7] did is done in this thesis. There are measures taken, how strong two politicians are related and the result is shown in a

[3] The Java Persistence API is the standard of OR-mapping in the Java world and Hibernate is an implementation of it.

network graph, in which one can visually see the groups of politicians which belong together and how strong the are related, either positively or negatively. Fortunately, all the speeches in the Austrian parliament are tagged with an annotation. A majority of them are tagged with either pro or con, speeches with other annotations are less interesting in the context of this thesis and won't be considered in the following analysis. The pro-con annotations of the speeches are used to create the relation graphs of all the politicians based on their position (pro or con) on the topics of the discussions they held speeches in. The resulting graph shows visually which relation two politicians have. If two politicians have a strong positive relationship (if they have the same attitudes in the discussions) they will be displayed close together, but if they have a strong negative relationship (mainly contrary attitudes) they will be displayed far away from each other.

The nodes of this graph are the politicians and the edges are the relationships between them. To create a good network graph, we have to measure the edge weight. Lucioni [7] used the voting data, whereas in the context of the Austrian thesis the speech annotations are used. The strength of a relationship (the link weight $w_p(p_1, p_2)$) between two politicians p_1 and p_2 is calculated via the following formula:

$$w_p(p_1, p_2) = \frac{\sum\limits_{i=0}^{D} w_p(d_i, p_1, p_2)}{\sum\limits_{i=0}^{D} |w_p(d_i, p_1, p_2)|}$$

where D is the total number of discussions and d_i is the i^{th} discussion. The result $w_p(p_1, p_2)$ is a normalized real number between -1 and $+1$, where $+1$ means that the two politicians had the same opinion on every topic and -1 means that they have totally contrary opinions. $w_p(d_i, p_1, p_2)$ is the weight of the politicians p_1 and p_2 in the discussion d_i and can be described by the following formula:

$$w_p(d_i, p_1, p_2) = \begin{cases} +1 & \text{if } p_1 \text{ and } p_2 \text{ have the same attitude in } d_i \\ -1 & \text{if } p_1 \text{ and } p_2 \text{ have contrary attitudes in } d_i \\ 0 & \text{if } p_1 \text{ or } p_2 \text{ did not speak in } d_i \end{cases}$$

In this thesis politician relation graphs get built for all legislative periods, to be able to view the structure of each period separately. To achieve to get a graph for only one period, only the discussions of this period were taken into account. There also has to be considered that there exists an edge between two politicians only if they spoke at least once in the same discussion.

3.5.2 Parliamentary Club Relation Graph

Similar graphs can be constructed for the parliamentary clubs. These graphs show how the clubs are related to each other. The weights of the edges of these graphs are computed by summing all weights of the politicians of the clubs and normalizing the sum. This can be expressed using the following formula:

$$w_c(c_1, c_2) = \frac{\sum\limits_{i=0}^{D} w_c(d_i, c_1, c_2)}{\sum\limits_{i=0}^{D} |w_c(d_i, c_1, c_2)|}$$

$$w_c(d_i, c_1, c_2) = \frac{\sum\limits_{i=0}^{P_{c_1}} \sum\limits_{j=0}^{P_{c_2}} w_p(d_i; p_{c_1,i}; p_{c_2,j})}{\sum\limits_{i=0}^{P_{c_1}} \sum\limits_{j=0}^{P_{c_2}} |w_p(d_i; p_{c_1,i}; p_{c_2,j})|}$$

The result $w_c(c_1, c_2)$ is again a normalized number from -1 to $+1$ and describes how the clubs are related. P_{c_x} describes the count of the politicians which belong to c_x and $p_{c_x,y}$ is the y^{th} politician which belongs to the parliamentary club c_x. Analogously to $w_p(d_i, p_1, p_2)$, $w_c(d_i, c_1, c_2)$ describes the weight of the clubs c_1 and c_2 in the discussion d_i and is also in the range of -1 to $+1$.

3.5.3 Pre-Calculation of Relation Weights

The calculation of all weights for the relations is quite expensive, as there are $\frac{n*(n-1)}{2}$ weights which have to be calculated for every discussion (n is the number of politicians which spoke in the discussion). So, if a graph should be displayed, it would take too long to compute all these weights then. To handle this issue, all politician relation weights $w_p(d_i, p_1, p_2)$ are calculated and persisted into the database immediately after new data is available (after the loading step of the ETL Application). When the data is needed to show the graph the database gets queried and using the aggregation functions of the database $w_p(p_1, p_2)$ and $w_c(c_1, c_2)$ can be easily derived for each pair of politicians and parliamentary clubs in an arbitrary period of time.

3.5.4 Other Analysis with the Relation Weights

With the now available weights between the politicians and parliamentary clubs also analysis other than the relation graphs can be applied. By summing and normalizing all the weights where one politician is in government and the other one is in opposition, a general tendency on how distinct government and opposition are can be calculated. See the results in section 4.3. Furthermore, the inner cohesion of government and coalition can be calculated by summing and normalizing all weights of politicians within these two groups. Another measure which can be taken out of the relation weights is the affinity of a politician to a certain party by again summing and normalizing all the weights of the politician and all the politicians of the party. Also the most and least related politicians of one selected politician can be derived by sorting all the calculated weights $w_p(p_1, p_2)$.

To-Do: soll ich hier auch formeln dazu angeben? würden halt alle ca so aussehen wie die von den Parlamentsklubs

3.6 Visualization

The visualization of the extracted data was an important task because only if the data is shown in a proper way it will potentially address a broad spectrum of people. That's why there was a big empathize on the data representation. Another requirement was that the application should run on as many devices as possible. Therefore a web application was implemented, as it will be accessible for almost anybody in the world, as long as he/she has a device with internet and a browser. A few web frameworks were evaluated and it was chosen an architecture with Spring Boot in the back end offering JSON/REST web services and Polymer in the front end for the visual representation.

3.6.1 Navigation Concept

Figure 3.3 shows the navigation concept of the implemented prototype. The start page shows an overview of the available pages and short statistics for the current legislative period like the mandate distribution and the overall absence. From the start page it

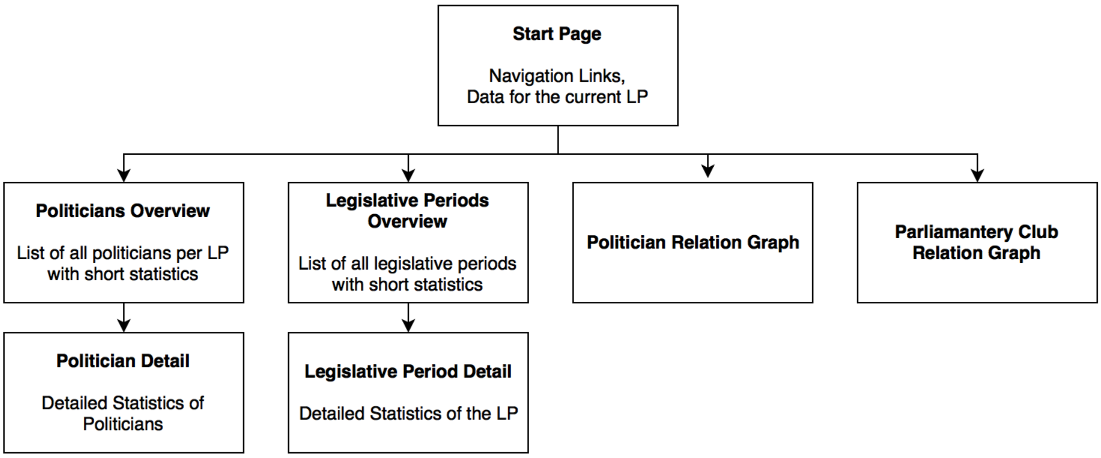

Start Page
Navigation Links,
Data for the current LP
Politicians Overview
List of all politicians per LP with short statistics
Legislative Periods Overview
List of all legislative periods with short statistics
Politician Relation Graph
Parliamantery Club Relation Graph
Politician Detail
Detailed Statistics of Politicians
Legislative Period Detail
Detailed Statistics of the LP

Figure 3.3: Navigation Concept of the Web Application

is possible to navigate to the politician overview, the legislative period overview, the politician and the parliamentary club relation graph pages.

The politician overview shows the members of the national council of a period, which can be specified. For each politician, the absence percentage and the speech count in this period are presented. If a politician is selected the politician's detail page gets opened. There is general data of the politician, the political mandates and the most/least related politicians shown.

The legislative period overview shows all legislative periods which were considered for this thesis (all from the 20^{th}) with short statistics. If a period gets selected, the period detail page where also a ranking of the most active and most absent politicians and clubs is shown.

On the politician and club relation graph pages there are graphs for the selected periods. In the politician relation graph there is also the possibility to filter for a specific topic of a discussion. Furthermore, a minimum count of speeches two politicians have spoken in common, before their relation is shown in the graph can be set. This creates the possibility to have a cleaner graph which only shows relations which are known to be correct until a certain degree.

3.6.2 Polymer

Polymer was chosen to implement the client side web application. The web framework was founded and developed by Google and enables web developers to create reusable web components (using JavaScript, HTML and CSS) which can be embedded in any HTML-Page. Furthermore, it has several built in functions which make it easier to create maintainable web applications using data binding and libraries for often used functionality like calling REST services. What's more, Polymer comes with several well designed components which make it easy to style a web application with the material design pattern from Google [4].

3.6.3 Relation Graphs

The relation graphs[4] were visualized using force layout design graphs of the d3 JavaScript library. As Bostock [2] describes in his paper about d3, the force layout is built with an algorithm proposed by Dwyer [3], which combines physical simulation and iterative constraint relaxation to show graphs with edges with different desired length.

The nodes of the graphs are the politicians/clubs and the edges were the relations between them. The more positive the relationship between two nodes is the closer they appear together in the graph. The relationship between two nodes is described with the weights w_p and w_c, respectively (see sections 3.5.1 and 3.5.2). So if the weight of the edge is $+1$ the desired length of the edge would be zero, whereas if the weight would be -1, the desired length would be a predefined maximum value (the minimum of graph width and height is used in the prototype).

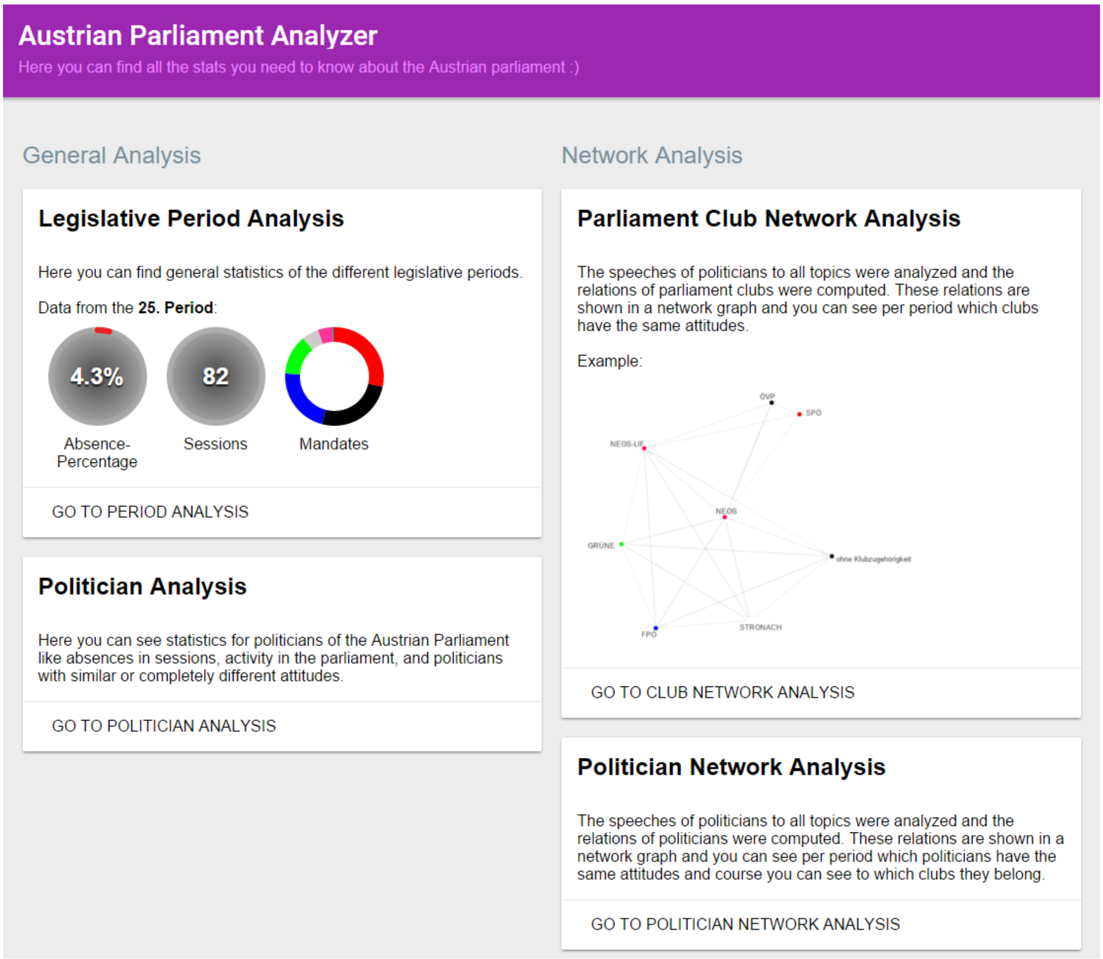
The calculated graph data is given into the algorithm of the d3 force layout graph and the algorithm tries to show a graph with the desired length of each edge. It improves the result iterative, this means that the graph looks better and better the more iterations were walked through. The result of the algorithm is then a more or less stable graph, where strong positive related nodes appear close to each other and negative related nodes appear far away from each other. This gives the possibility to identify communities of politicians which have similar attitudes and opinions.

[4] See sections 3.5.1 and 3.5.2 for the creation of the graphs and chapter 4 for the resulting graphs.

Chapter 4

Results and Discussion

A web application was developed which shows the results of the analysis applied on the data which got extracted and makes them accessible for everybody. Figure 4.1 shows the start page of this application.



Austrian Parliament Analyzer

Here you can find all the stats you need to know about the Austrian parliament :)

General Analysis

Legislative Period Analysis

Here you can find general statistics of the different legislative periods.

Data from the **25. Period**:

GO TO PERIOD ANALYSIS

Politician Analysis

Here you can see statistics for politicians of the Austrian Parliament like absences in sessions, activity in the parliament, and politicians with similar or completely different attitudes.

GO TO POLITICIAN ANALYSIS

Network Analysis

Parliament Club Network Analysis

The speeches of politicians to all topics were analyzed and the relations of parliament clubs were computed. These relations are shown in a network graph and you can see per period which clubs have the same attitudes.

Example:

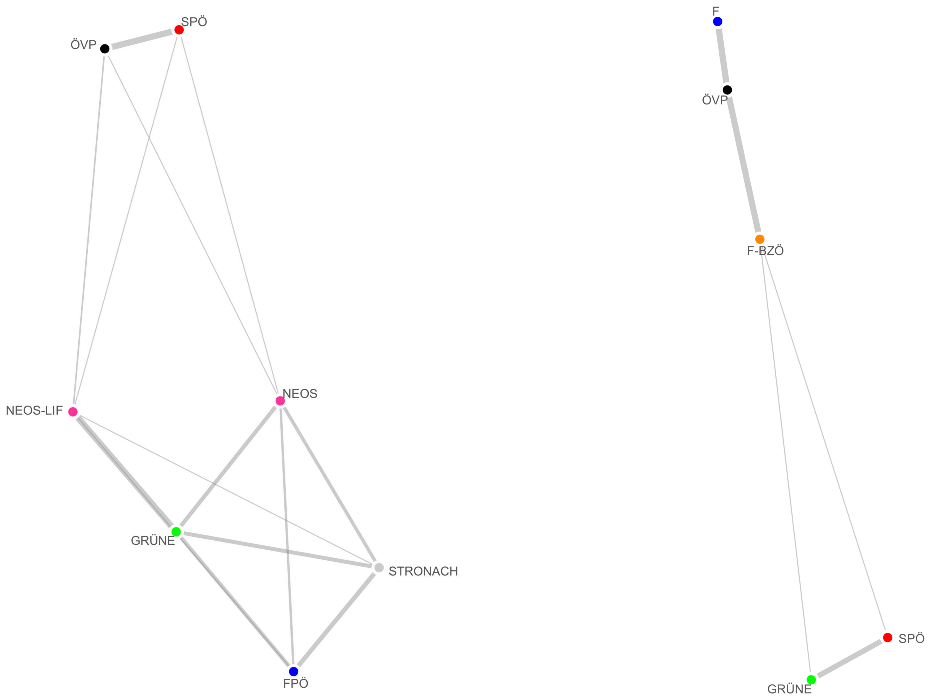
GO TO CLUB NETWORK ANALYSIS

Politician Network Analysis

The speeches of politicians to all topics were analyzed and the relations of politicians were computed. These relations are shown in a network graph and you can see per period which politicians have the same attitudes and course you can see to which clubs they belong.

GO TO POLITICIAN NETWORK ANALYSIS

Figure 4.1: Start Page of the Prototype Web Application



(a) 25. Period

(c) 22.Period

Figure 4.2: Club Relation Graph

4.1 Relations of Parliament Clubs

Figure 4.2 shows the created relation graphs of the 25. and 22. legislative periods. The more positive the relation between two clubs is, the closer they appear together in the graph and the thicker is the edge between them. Only positive edges are shown in the graph, as the graph would be too confusing if the negative edges were also shown. In all graphs shown, one can easily see that there are two groups of parliamentary clubs in the parliament: Those which are in government and those which are in opposition. In table 4.1 the parties are listed per period weather they were in government or in opposition.

Legislative Period	Governing Parties	Opposition
20. Period: 1995 - 1999	SPÖ, ÖVP	FPÖ, Grüne, Liberale
21. Period: 1999 - 2002	ÖVP, FPÖ	SPÖ, Grüne
22. Period: 2002 - 2006	ÖVP, FPÖ, BZÖ[1]	SPÖ, Grüne
23. Period: 2006 - 2008	SPÖ, ÖVP	FPÖ, Grüne, BZÖ
24. Period: 2008 - 2013	SPÖ, ÖVP	FPÖ, Grüne, Stronach, BZÖ
25. Period: since 2013	SPÖ, ÖVP	FPÖ, Grüne, NEOS, Stronach

Table 4.1: Government and Opposition in the Legislative Periods 20 to 25

4.2 Relations of Politicians

Graph + Explanations

4.3 Government - Opposition Relation

Legislative Period	Government-Opposition Relation	Inner Government Relation	Inner Opposition Relation
20. Period	-0.85	+1.00	+0.86
21. Period	-0.695	+0.985	+0.908
22. Period	-0.567	+1.00	+0.938
23. Period	-0.382	+0.994	+0.765
24. Period	-0.52	+1.00	+0.768
25. Period	-0.374	+1.00	+0.676

Table 4.2: Government-Opposition Relation, Inner Government- and Inner Opposition Relation for the Legislative Periods 20 to 25

[1] The BZÖ was in government from 17^{th} of April, 2005

Chapter 5

Conclusions and Future Work

The implemented prototype shows that it is possible to get structured data out of the stenographic transcripts and that analysis can be performed on the basis of this data. The analysis can give an overview of politician absence, length of serving in the national council, politician activity, the relation between politicians/clubs and the inner cohesion of coalition or opposition.

Bibliography

[1] A. Amelio and C. Pizzuti. Analyzing voting behavior in italian parliament: Group cohesion and evolution. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 140–146, Aug 2012.

[2] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[3] Tim Dwyer. Scalable, versatile and simple constrained graph layout. In *Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'09, pages 991–1006, Chichester, UK, 2009. The Eurographs Association &#38; John Wiley &#38; Sons, Ltd.

[4] Google Inc. *Polymer Project*, 2015. `https://www.polymer-project.org/1.0/`.

[5] Forum Informationsfreiheit. *Informationsfreiheit.at*, Dec 2015. `http://www.informationsfreiheit.at/`.

[6] Forum Informationsfreiheit. *OffenesParlament.at*, Dec 2015. `http://offenesparlament.at/`.

[7] Renzo Lucioni. *Senate Voting Relationships*, 2013. `http://www.renzolucioni.com/senate-voting-relationships/`.

[8] Austrian Parliament. *The Austrian Parliament*, Dec 2015. `http://www.parlament.gv.at/ENGL/PERK/PARL/`.

[9] Mason A. Porter, M. E. J. Newman, Peter J. Mucha, and Casey M. Warmbrand. A network analysis of committees in the u.s. house of representatives. *PNAS*, Jan 2005.