TECHNICAL UNIVERSITY MUNICH

TUM Data Innovation Lab

A Network Analytical take on the European Parliament

Authors: Abinav Ravi Venkatakrishnan
Niklas Schmidt
Mentor: Dr. Mirco Schönfeld (Professorship for Computational Social Science & Big Data)
Co-Mentor: Laure Vuaille (Department of Mathematics)
Project Lead: Dr. Ricardo Acevedo Cabra (Department of Mathematics)
Supervisor: Prof. Dr. Massimo Fornasier (Department of Mathematics)

Contents

1 Introduction

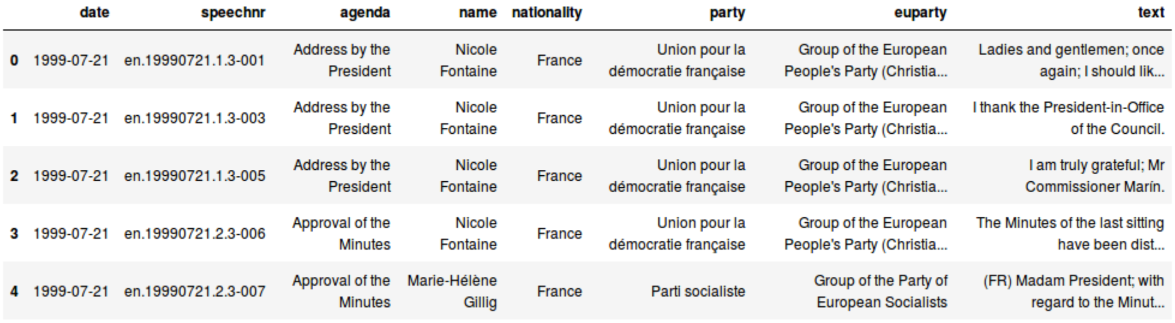
Network Analysis is a very effective way of modeling a complex interactive structure of the European parliament. Network Analysis is primarily done so as to provide interpretation for importance, influence and control of Members of the European Parliament (MEP) in the network. This is important as we can identify and track over multiple parts of the network at the same time.

The main goal of the project was to find hidden agendas and quantify the influence of MEPs on the hidden agenda. The initial problems faced were that we had to define what can be considered as a hidden agenda as it can be described in multiple ways. In this project we have worked with 2 such interpretations one of which we called hidden coalition and have two methods of network analysis.

The report is structured as follows. Section 2 of the report talks about the dataset and the extraction of data. Section 3 talks about the topic modeling approach and extracting the optimal number of topics from the given corpora. Section 4 is about how to build a network with topic modeling and conduct the analysis for the research question. Section 5 and Section 6 talks about inference and future work.

2 Dataset

The data used for this project were the official minutes of the European Parliament (EP) which are available at the EP's website. However since there is no easy way to get a clean, machine-readable dataset from there we used the RDF-database at http://linkedpolitics.ops.few.vu.nl, which has all the speeches from July 1999 until July 2017 stored together with corresponding metadata. Thus we could easily access all the relevant data by simple SPARQL queries. Our final dataset consisted of 132,356 entries containing the translated speeches until 2012 (the EP stopped translating speeches that year) together with information about the speaker (name, nationality, national and european party) as well as the date and official agenda belonging to the speech.



	date	speechnr	agenda	name	nationality	party	euparty	text
0	1999-07-21	en.19990721.1.3-001	Address by the President	Nicole Fontaine	France	Union pour la démocratie française	Group of the European People's Party (Christia...	Ladies and gentlemen; once again; I should lik...
1	1999-07-21	en.19990721.1.3-003	Address by the President	Nicole Fontaine	France	Union pour la démocratie française	Group of the European People's Party (Christia...	I thank the President-in-Office of the Council.
2	1999-07-21	en.19990721.1.3-005	Address by the President	Nicole Fontaine	France	Union pour la démocratie française	Group of the European People's Party (Christia...	I am truly grateful; Mr Commissioner Marín.
3	1999-07-21	en.19990721.2.3-006	Approval of the Minutes	Nicole Fontaine	France	Union pour la démocratie française	Group of the European People's Party (Christia...	The Minutes of the last sitting have been dist...
4	1999-07-21	en.19990721.2.3-007	Approval of the Minutes	Marie-Hélène Gillig	France	Parti socialiste	Group of the Party of European Socialists	(FR) Madam President; with regard to the Minut...

Figure 1: Excerpt of the dataset

2.1 Data Preprocessing

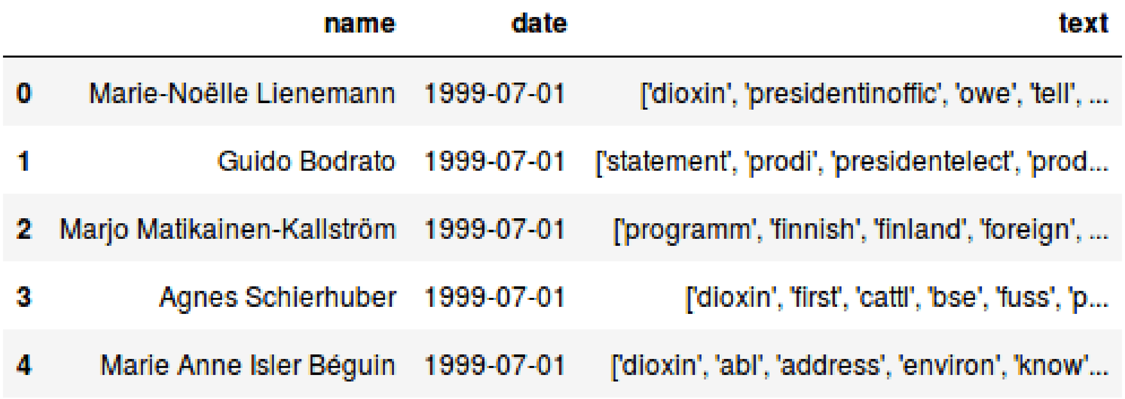
For the Topic Modelling described in the next chapter we had to bring the plain text of the speeches into a feasible format as well as removing texts that can not actually be classified as a speech, for instance "I thank the President-in-Office of the Council.". Despite finding information that heavy preprocessing might even lead to worse results in the Topic Modelling we found that sparse preprocessing as described in [1] did not provide reasonable results. In the end we performed the following steps:

1. Concatenation of agenda and speech to support the Topic Modelling.
2. Removal of punctuation and lowercasing of the whole text.
3. Tokenization the text.
4. Removal of stopwords.
5. Removal of most common words appearing in the whole text corpus.
6. Lemmatizing and stemming of the remaining words.
7. Removal of "fill words", that is words for addressing the audience and organizational matters like "ladies", "comission" or "question", and removal of words consisting of less than 3 letters.
8. Removal of texts with less than 10 words remaining

The last step was done to reduce the amount of texts not being classified as speeches as discussed above. As there was quite a big variation in the length and type of the speeches we were not able to accurately remove all the other remarks and instead used this quite reasonable threshold appporach. Finally for all MEPs we aggregated all their speeches of one session (one month) due to several reasons. Firstly speeches were sometimes split up into different entries in our dataset, possibly by non-recorded interruptions (see first lines of Figure 1). Secondly it made sense to see one session as one entity where the topics of one MEP do not vary much. And lastly not aggregating the speeches would mean assigning people with more (and hence maybe shorter) speeches more influence in the final analysis which will become clearer in the following chapters. However speaking about the same topic multiple times across multiple sessions should be weighted more as it represents dedication to this topic. With these steps we were able to get good and stable results for the Topic Model.

3 Topic Modelling

Topic Modeling is a text mining method that can be used for Dimensionality reduction and for analysis of large scale Data Analysis. Most of the Topic Modeling methods are based on the maximum likelihood estimate of the underlying probability distributions.



	name	date	text
0	Marie-Noëlle Lienemann	1999-07-01	['dioxin', 'presidentinoffic', 'owe', 'tell', ...
1	Guido Bodrato	1999-07-01	['statement', 'prodi', 'presidentelect', 'prod...
2	Marjo Matikainen-Kallström	1999-07-01	['programm', 'finnish', 'finland', 'foreign', ...
3	Agnes Schierhuber	1999-07-01	['dioxin', 'first', 'cattl', 'bse', 'fuss', 'p...
4	Marie Anne Isler Béguin	1999-07-01	['dioxin', 'abl', 'address', 'environ', 'know'...

Figure 2: Preprocessed and aggregated dataset

Topic modeling is also used to learn about the thematic structure from large collection of documents.

One of the main importance of topic models is the patterns of word use and connect documents that share similar patterns. Documents are viewed as a mixture of topics. The mixture of topics is generally viewed as a mixture of probability distribution over words in a document. Most of the topic modeling techniques in literature use a bag of words technique which essentially ignores the information of ordering of words.

3.1 Latent Semantic Analysis

Two topic modeling techniques extremely popular are the Latent Semantic Analysis(LSA) and Latent Dirichlet Allocation(LDA). Latent Semantic Analysis [2] is a method that creates a vector based representation of texts to make semantic content. It is done by checking similarity between texts and picking the most efficient one. LSA uses Single value Decomposition to get the correlation of topics with probabilities. But the disadvantages of LDA according to a paper [3] are that there are lack of interpretable embeddings. The computation requires large set of documents and vocabulary. It is also very less efficient.

3.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation [4] is the method that is used to do Topic Modeling on the data.LDA can be thought of as a process that tends to mimic the way that documents are written. It is better than LSA in a way that it represents mixture models that captures exchange-ability of words and documents.

LDA is a Bayesian analysis which is generative in nature. Each document is modeled as a mixture of topics and each topic as a mixture of probability of words. Figure 3 shows a plate modeling of LDA. Topics and words are chosen from the Multinomial distribution

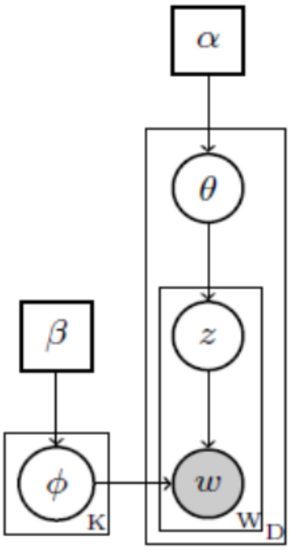


Figure 3: Plate model of the LDA

and the priors are taken to be Dirichlet prior. The term dirichlet prior tells us that the priors are taken over pdf of distribution itself.
We can say that the LDA is different from multinomial clustering by the following aspects. Clustering is a 2 level model where a Dirichlet is a sampled from corpus and multinomial clustering variable is also sampled from corpus and the set of words are conditional on cluster variable whereas LDA is 3 level model which is sampled repeatedly with the document for topics. which allows a particular document to be asssociated with multiple topics.
The inference is done on a basis of Expectation Maximization Algorithm where in the E step the optimizing of Variational parameters of the distribution are done and in the M step Maximum Likelihood estimates are found with appropriate posterior from E step.
We use a library called gensim for modeling our Text corpora.

3.3 Choosing Optimal Topics:

Since Topic modeling is an Unsupervised learning method the latent parameter of number of topics is unknown. There are certain automatic metrics that have been mentioned in Mimno.et.al [5] which mentions topic size and topic coherence as metrics. But they also mention that Topic size is bad since bad topics have shorter number of words in them than good oones which makes domain specific topic modeling a little difficult.

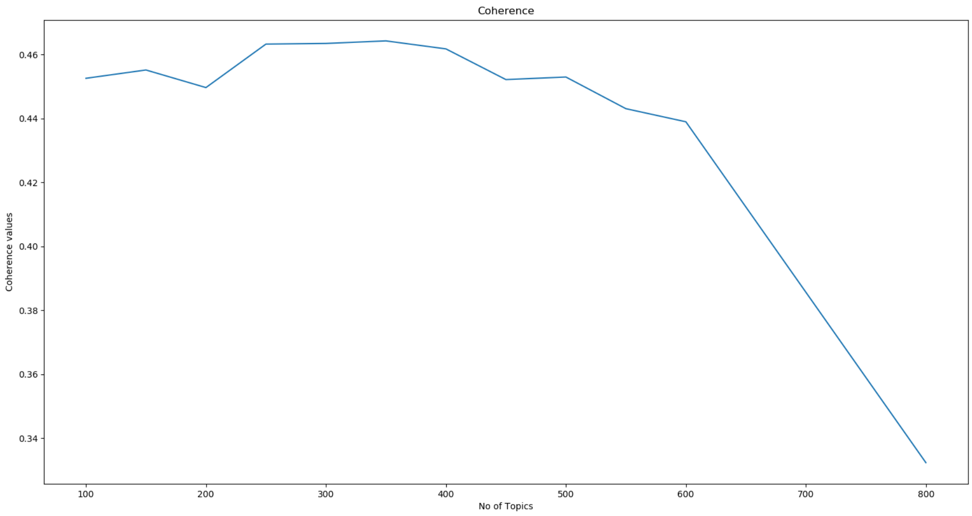


Figure 4: Coherence values for given number of topics

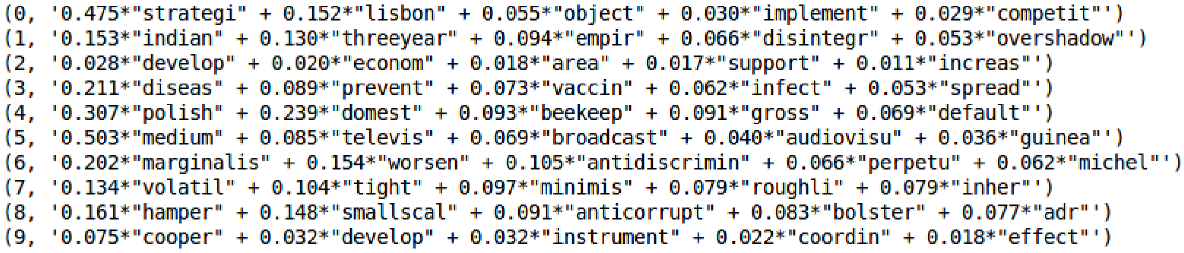
Topic coherence is defined as

$$C(t, V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(V_m^{(t)}, V_l^{(t)}) + 1}{D(V_l^{(t)})}$$

where V^t is a list of M most probable words in the topic t.
D(v) is the document frequency of word type v and D(v,v') be the co-document frequency of word types v and v' i.e. the number of documents containing one or more tokens of type v and atleast one token of type v'.
Topic Coherence has better precision and Area under the curve (AUC) values when compared with topic size as a metric.
We used the function Coherence function in gensim and did a grid search on the number of topics. The topic with the highest coherence value was selected and further analysis was done on that. The number of topics is 350 for our data corpus. We can see in Figure 4 about the coherence values starting from 100 topics to 1000 topics.

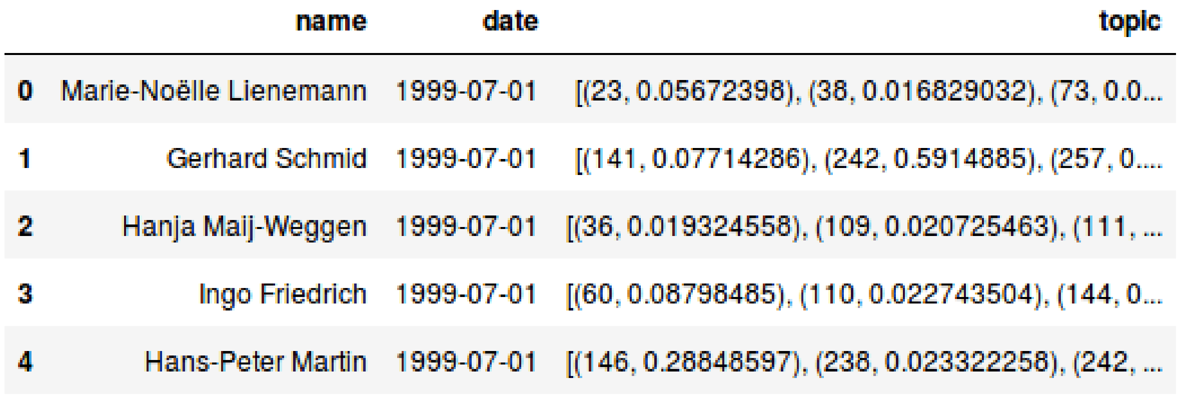
3.4 Inference of Topic Model

From the topic modeling we get our output as a list of words with certain probabilities. Now we infer that a person in a particular session can be associated with different topics. So the MEP is assigned with the corresponding topic numbers and their probabilities as a part of the session. Figure 6 gives an example of inferred topics per session for the MEPs.



```
(0, '0.475*"strategi" + 0.152*"lisbon" + 0.055*"object" + 0.030*"implement" + 0.029*"competit"')
(1, '0.153*"indian" + 0.130*"threeyear" + 0.094*"empir" + 0.066*"disintegr" + 0.053*"overshadow"')
(2, '0.028*"develop" + 0.020*"econom" + 0.018*"area" + 0.017*"support" + 0.011*"increas"')
(3, '0.211*"diseas" + 0.089*"prevent" + 0.073*"vaccin" + 0.062*"infect" + 0.053*"spread"')
(4, '0.307*"polish" + 0.239*"domest" + 0.093*"beekeep" + 0.091*"gross" + 0.069*"default"')
(5, '0.503*"medium" + 0.085*"televis" + 0.069*"broadcast" + 0.040*"audiovisu" + 0.036*"guinea"')
(6, '0.202*"marginalis" + 0.154*"worsen" + 0.105*"antidiscrimin" + 0.066*"perpetu" + 0.062*"michel"')
(7, '0.134*"volatil" + 0.104*"tight" + 0.097*"minimis" + 0.079*"roughli" + 0.079*"inher"')
(8, '0.161*"hamper" + 0.148*"smallscal" + 0.091*"anticorrupt" + 0.083*"bolster" + 0.077*"adr"')
(9, '0.075*"cooper" + 0.032*"develop" + 0.032*"instrument" + 0.022*"coordin" + 0.018*"effect"')
```

Figure 5: Excerpt of the topics defined by a set words. Here the 5 most descriptive words are displayed.



	name	date	topic
0	Marie-Noëlle Lienemann	1999-07-01	[(23, 0.05672398), (38, 0.016829032), (73, 0.0...
1	Gerhard Schmid	1999-07-01	[(141, 0.07714286), (242, 0.5914885), (257, 0....
2	Hanja Maij-Weggen	1999-07-01	[(36, 0.019324558), (109, 0.020725463), (111, ...
3	Ingo Friedrich	1999-07-01	[(60, 0.08798485), (110, 0.022743504), (144, 0...
4	Hans-Peter Martin	1999-07-01	[(146, 0.28848597), (238, 0.023322258), (242, ...

Figure 6: Inferred topics per MEP and session

4 Network Analysis

4.1 Network Modelling

After training our Topic Model and inferring topic probabilities to the speeches our next task was to build a network that we could analyze. Building this network consisted of two steps:

1. Building a bipartite 2-mode network with topics and MEPs as nodes.
2. Folding the 2-mode network resulting in a network with only MEPs as nodes.

1.
We build the 2-mode network as follows:
We start by restricting the dataset to the time period we want to observe. Then every entry in the dataset has an MEP and a corresponding list of topics and probabilities (cf. Figure 6). For each topic in this list add an edge between the MEP-node and the topic-node (creating these nodes if they do not already exist) with the probability as the edge weight. In case this edge already existed add the probability to the edge weight instead of creating a new edge. Repeat this procedure for all entries in the dataset.
2.
The 1-mode network is constructed from the 2-mode network as follows:

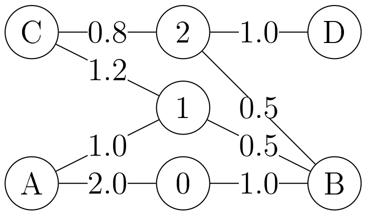
For every pair of MEP-nodes that are connected to at least one same topic-node there will be an edge between them in the 1-mode network. For every topic-node through which they were connected we multiply the corresponding edge weights. Summing up these multiplied values gives the edge weight in the 1-mode network. Additionally we store a vector with the edge weights between the MEP-node and the respective topic-nodes in the MEP-nodes which we will call topic data. Note that the edge weight between two nodes is exactly the standard scalar product of the corresponding topic data vectors.

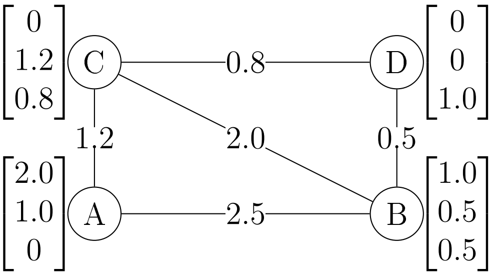
Example:

We visualize this construction with a small toy example. Let the data after the inference of the topic model be as follows:

name	**date**	**topic**
A	1999-07-01	[(0,0.7), (1,0.3)]
B	1999-07-01	[(0,0.7), (1,0.3)]
C	1999-07-01	[(1,0.7), (2,0.3)]
D	1999-07-01	[(2,1.0)]
A	1999-08-01	[(0,0.5), (1,0.5)]
B	1999-08-01	[(0,0.3), (1,0.2), (2,0.5)]
C	1999-08-01	[(1,0.5), (2,0.5)]
A	1999-09-01	[(0,0.8), (1,0.2)]

This gives the following 2-mode network (left) and finally the 1-mode network (right):





4.2 Analysis

Before we could analyze our networks we had to first specify our goals. We had to find a definition for a hidden agenda. We ended up with the two following similar concepts:

Hidden Agenda: An MEP follows a hidden agenda if they are trying to achieve some goal in a non-obvious manner, instead of for example advocating for it through their speeches.

Hidden Coalition: A pair of MEPs have a hidden coalition if their connection/ collaboration is not apparent by their direct work on the same subjects.

For both definitions we developed a different approach. Both build upon the idea of community detection in networks and use the so called Louvain algorithm proposed in [6]. This is a greedy algorithm which tries to maximize the modularity of the network which is defined as the ratio between the number of intra-community edges minus the expected ratio if the edges were randomly distributed preserving degree distribution.

Let $G = (V, E)$ be a graph with vertices V and edges E. Let $c_v \in \{1, \dots, K\}$ be the community of $v \in V, a_{vu}$ the weight of the edge between v and $u, k_v = \sum_{u \in V} a_{vu}$ and $m = \frac{1}{2} \sum_{v \in V} k_v$. Then the *modularity* of the community partition is given by

$$Q = \frac{1}{2m} \sum_{v,u \in V} \left(a_{vu} - \frac{k_v k_u}{2m} \right) \delta(c_v, c_u)$$

where δ is the Kronecker delta.

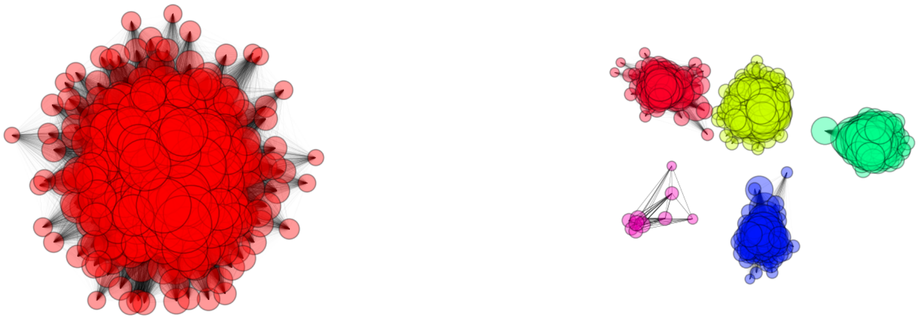


Figure 7: Original graph and result of the Louvain algorithm (with inter-community edges removed).

4.2.1 Community outlier

The basic idea of this approach is to find some global structure in our network and observe MEPs that do not fit this structure locally classifying them as outliers. Recurring outliers over different time peridos would then be considered candidates for MEPs following a hidden agenda. In more detail this approach builds upon the idea of the Girvan-Newman algorithm for community detection [7]. To explain this algorithm we also need the follwoing definition.

Let $G = (V, E)$ be a graph with vertices V and edges E. For $e \in E$ we define the *edge betweenness centrality* as

$$c(e) = |\{(v, u) \in V^2 : e \text{ belongs to shortest path from } v \text{ to } u\}|.$$

The algorithm now iteratively removes the edge with the highest betweenness centrality until no edges are remaining. In each iteration the communities are given by the connected components of the remaining graph. For the final community partition we chose the one with the highest modularity among all those appearing during execution of the algorithm.

Due to the construction of our network we will have larger edge weights whenever the MEPs of the adjacent nodes have the same topics in common that they talked about. As a consequence these edges are less likely to be in a shortest path as we are minimizing path costs. Thus MEPs with the same topics will more likely be in the same community. On the other hand MEPs with fewer common topics will have a smaller weight and thus be more likely to end up in different communities. Thus we can assume that the topic distribution of the nodes is quite similar in one community. We now call an MEP an outlier if their topic data deviates too much from the mean of the community. Having such an outlier means that the connection between them and the community must be strong enough to keep at least one edge despite having non-fitting data. This in turn means that their overall topics were still close enough to that community without it being obvious which fits our definition of a hidden agenda.

To quantify this deviation we had a couple of different approaches. Our first idea was instead of using the node data directly to use the edge data given by entrywise multiplication of the topic vectors of the adjacent nodes. Then remove all edges where the value of the hottest topic of the edge data, that is the topic with the highest mean value, lies below a certain threshold, for example the corresponding mean, and calling all nodes outliers that get disconnected from the community by this procedure. Another idea was to quantify the deviation of the individual node data from the mean community data by measures like the cosine similarity. However in the second case we would have had to use some hard threshold and in the first case we did not take possible multiple hot topics into account.

In the end we used the following approach. We first observe how the topics of one community are distributed. As we can see in Figure 8 for the majority of the topics the values are accumulated close to 0 with a few outliers to the top. However there are a few topics where the mean value is much higher. We call an MEP outlier if there is a topic for which the value of the MEP is lower than the mean minus the standard deviation. Or mathematically precise:

Definition:

Let $G = (V, E)$ the network and T be the number of topics, $t \in \mathbb{R}^T$ the topic data of node $v \in V$, $m, s \in \mathbb{R}^T$ the topic mean values and standard deviations of the community c_v respectively. Then v is an *outlier* if there is some $1 \leq i \leq T$ such that $t_i < m_i - s_i$.

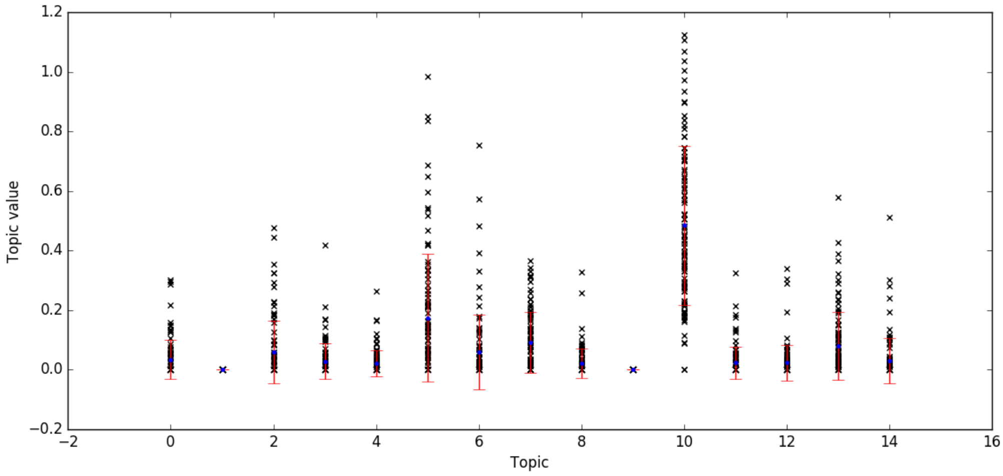


Figure 8: Topic distribution of one community (The weights were scaled to lie beween 0 and 1) together with mean values (blue) and standard deviation (red). For clarity we used a topic model with only 15 models in Figures 8, 9 and 10, however the results are similar when using the optimal number of topics.

Remark:

1. By construction (adding weights if MEPs have multiple speeches) high values t_i are not unusual and thus will not be considered as outliers. Also having an unsual high value where the other values in this community are also high would result in large edge weights and we would thus expect the corresponding MEP to be in this community. However we will see that there is a certain connection between outliers and high values.

2. Note that we only require $t_i < m_i - s_i$ instead of $t_i < m_i - 2s_i$ how outliers are usually determined. As "outliers" to the top are relatively common the standard deviation is already quite large.

3. This final approach is similar to our first idea however we now take all the topics into account instead of just the hottest one. Also instead of using the edge data we work directly on the node topic data.

Allthough our approach is influenced by the Girvan-Newman algorithm we could not use it due to the runtime of the algorithm which was infeasible for our large, densely connected network. Instead as already described above we used the Louvain algorithm which has a significantly better runtime.
To gain an understanding of why outliers were still connected to their communities we compared the topic distribution of the outliers to the community. Our first observation was that often having a value below the standard deviation of the hot topics was compensated by having other values above the standard deviation - often for multiple topics

(cf. Figure 9). If we look at the direct neighbours of one outlier that remain in the community we can see that their distribution mostly follows the mean of the community whilst still having some overlaping topis with the outlier (cf. Figure 10).

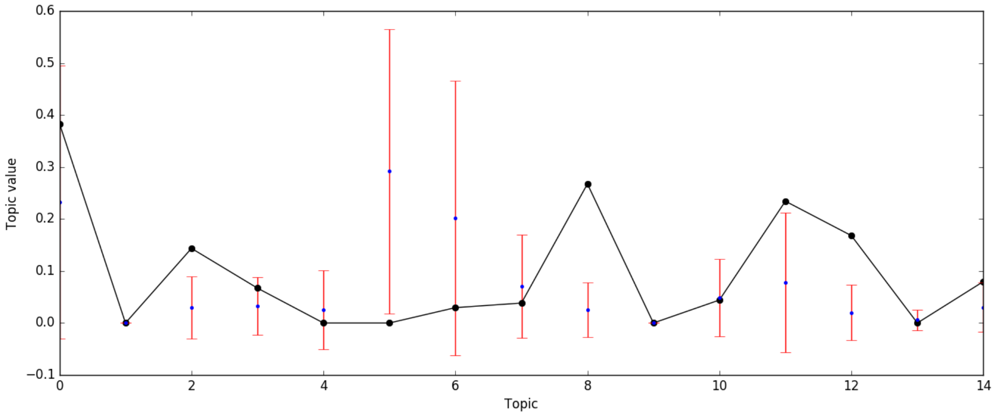


Figure 9: Distribution of the topics of one outlier (black) compared to the mean and standard deviation of its community

Now it might happen that MEPs get misclassified as outliers maybe because they were assigned to the "wrong" community in the beginning or because their overall contribution was just not enough resulting in low values across all topics. To counter this phenomenon we observe the outliers across multiple time periods of different length and only take recurring outliers as candidates for MEPs following a hidden agenda.

As our data was unlabelled and we designed the approach ourselfs we had to find some way to evaluate the quality of the approach. We decided to go for a qualitative approach and compare the speeches of the outliers to those of the rest of the community. We detected that outliers often talked about multiple different subjects while the speeches of the remaining mostly sticked to the same subject.

4.2.2 Hidden Communities

Networks contain a set of communities which can be called as dominant communities which interfere with the detection of weak, natural community structure. These weak communities are hard to discover since the members of the weak communities also belong to the dominant communities. These weak communities are known as hidden community structure.

This is shown by a paper published by he.et.al [8] For example we can consider that in a workplace people belonging to certain teams as a strong community since they are working together on a similar topic. There might be other groups such as athletics group, jazz group etc which may contain people from different teams. These communities tends to less modular than the original ones and hence is generally overlooked. We can notice the example of this in Figure 11.

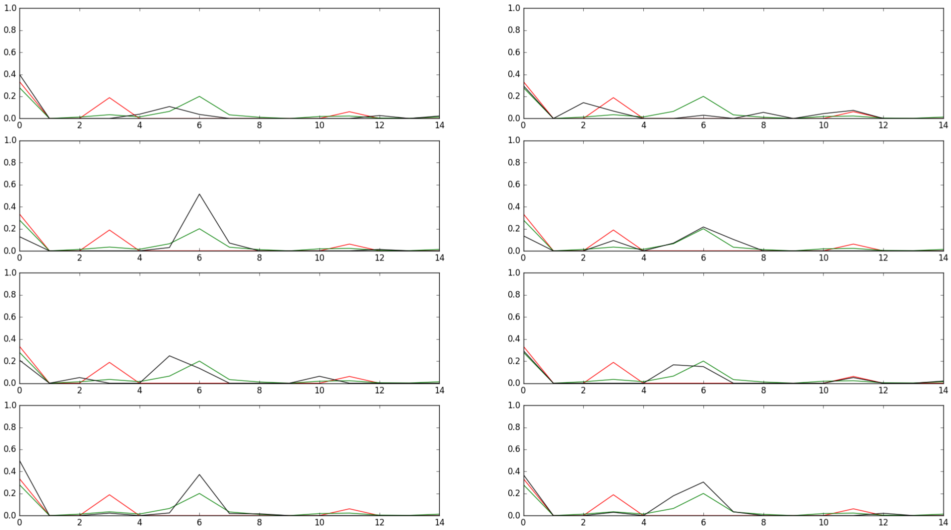


Figure 10: Comparison between one outlier (red) the community mean (green) and some of its direct (non-outlier) neighbours in the community

Preliminaries: Let graph $G = (V, E)$ represent a network with n nodes and e edges. Let A be the adjacency matrix of G, the ij entry $A_{ij} \in 0, 1$ indicates whether there is an edge connecting nodes i and j. C is a set of all the overlapping communities $C_1, C_2, ..., C_K$ all the communities are a sub-graph of the main graph G.
Algorithm: The algorithm of Hidden community detection mainly consists of 2 stages of Identification and refinement.In the identification stage we determine the initial layers of community as follows
Identification:

1. Identify a layer of communities via the base method.
2. Weaken the structure of the detected layers.
3. Repeat until the appropriate number of layers are found.

The next step is a set of refinement steps that need to be done on the identification. After identification an approximate of various community layers are found. This layer improves the quality of the layers found. Refinement step is done as follows
Refinement:

1. Weaken the structures of all other layers to obtain a reduced network
2. Apply the base algorithm to the resulting network.

Refinement Method There are refinement methods that are possible. They are

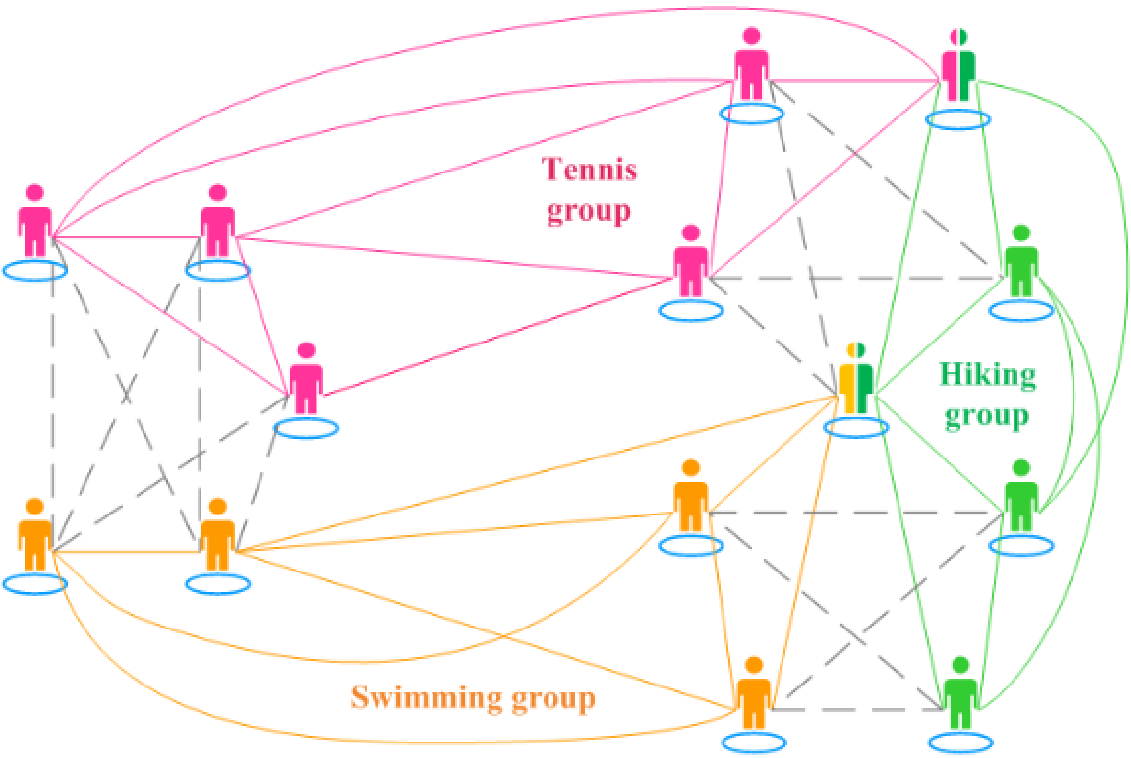


Figure 11: An example of how the Idea of Hidden community works from He.et.al [8]

1. **Remove Edge**: This method removes all the intra-community edges like the Girvan-Newman algorithm which removes edges with high betweenness centrality. Remove edge works well with small overlaps.

2. **ReduceEdge:** This method approximates each layer as a single stochastic block-model with other edges regarded as background noise. This method randomly removes edges with each community block so that the edge probability matches the background edge probability of the block.
   The observed edge probability in a community is given by

   $$p_k = \frac{e_{kk}}{0.5n_k(n_k - 1)}$$

   The outgoing edge density is given by

   $$q_k = \frac{d_k - 2e_{kk}}{n_k(n - n_k)}$$

   where

   e_{kk} - edges inside the community
   n_k - nodes in the community
   d_k - degree of the community

The observed edge probability in a community is treated as the superposition of the underlying edge probability. The probability that the edge is generated from background noise is 1 - observed edge probability.

Selection of Number of layers: A major challenge is deciding what is the number of layers upto which the algorithm must be applied. We observe that the average modularity increases during the refinement stage of the algorithm which implies that for the right amount of layers the quality of the output increases.
Algorithm for selecting number of layers:

1. Calculate Q_0 for $t = 0$ before any refinement is done

2. Perform T = 10 tentative iterations of refinement and calculate the Modularity Q_t for $t \in 1, ...T$

3. Calculate the average improvement ratio of modularity per iteration as

$$R_T = \frac{\sum_{t=1}^{T} Q_t}{T \cdot Q_0}$$

R_T represents how much refinement improves the layers. We choose the n_L corresponding to R_T.

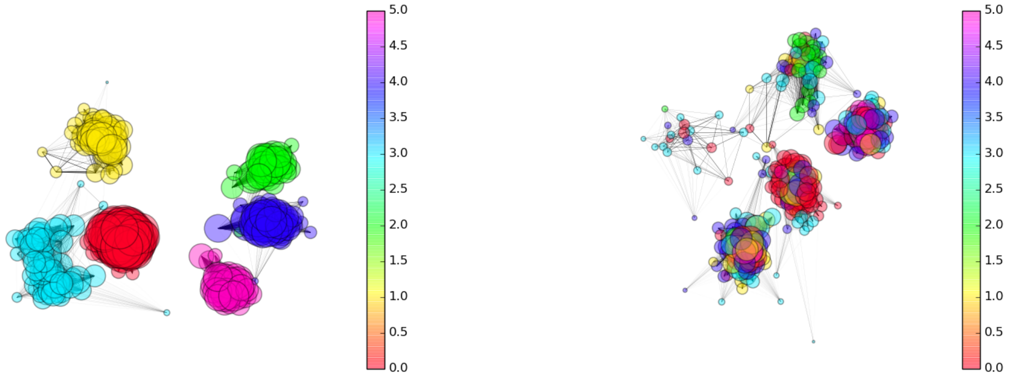


Figure 12: Dominant (left) and Hidden Communities (right). The labels/colors correspond in both plots to the dominant communities.

Finding Hidden Coalition: Hidden Coalition is found out by considering the pair of MEP who don't share a dominant community but occur quite often together in lower layers during the Hidden community detection. It loosely translates to that they are being identified with communities that they are less probable to belong. If their occurences are more then we can say that there is a hidden coalition between the pair of MEP being considered.

4.2.3 Failed appproach

There was one other approach we tried that seemed really promising and well-fitted for the hidden agenda detection namely the so called *Hollistic Community Outlier Detection* presented in [9]. This algorithm finds communities and outliers directly by making use of the network structure and data on the nodes and edges. The first problem we faced was that the edge data we used (entrywise multiplication of the adjacent node data) was by construction not independent of the node data. As we could not come up with a better construction for the edge data we tried using a reduced method that only takes the node data into account but this did not yield reasonable results. Thus we dropped this approach and focused on the other two. However we still think that with some more refinement this method can be used to find hidden agendas.

5 Results

Eventually we applied our approaches to the two full legislative periods in our dataset: 07/1999-06/2004 and 07/2004-06/2009. For each subset of 3 consecutive months we built and analyzed the corresponding network with our two approaches. We then counted for how many of these networks each MEP respectively MEP pair was detected by our algorithms and manually examined those occuring the most often.

Hidden Agenda:

For both terms we looked at the top 10 recurring outliers which corresponds to those being detected in more than 40% of the networks in the first legislative period and more than 27% in the second. Out of those a (vice) president or Secretary General position was held by 2 respectively 4 MEPs during the first and second term. This somehow makes sense as they guide their respective groups and have a superordinate role and by that being involved in many different subjects. So we have a closer look at the remaining candidates which give an interesting result. The majority of the rest (7 and 5) are members of left- and right-wing parties the most prominent example of the recent past probably being Marine Le Pen.

Hidden Coalition:

For both the legislative periods we looked into which pair of MEPs occur in the same community in the hidden layer for a large number of time and chose a probability of them occuring more than 20% of the time in the first session and 15% in the second session.

We then compared these MEP with the party and the topics that they were working on we found that out of the top 10 pair of MEP chosen by the above mentioned criteria about 60% spoke on various different topics and that's why they are in the same communities but there are pairs which don't speak of the same topics yet come in the similar communities quite often in the legislative period these people can be said to have a hidden coalition.

- First legislative session - 4/10 pairs can be said to have hidden coalition.

- Second legislative session - 4/10 pairs can be said to have a hidden coalition.

6 Conclusion

In this project we have explored ways of defining and finding out hidden agenda. We have extracted the data from a graph database, done topic modeling with coherence as our prime metric to optimize the number of topic models. We have then modeled the network by using the topic modeling data and have proposed two different methods of finding hidden agenda and hidden coalition. The hidden agenda method uses an outlier detection algorithm that tracks the hottest topic in the community and checks which MEP has the least probability to speak about and propose that if the MEP is an outlier he might be having hidden agenda. The second approach takes a hierarchical approach and finds the non domiinant communities and MEP who are not in the first layer but interact in the lower layers are said to have a hidden coalition.

However there were some things we were not able to do due to the time limit which could be interesting for future work. The first was already mentioned in 4.2.3 namely the refinement of the HCOutlier detection. Then we originally planned to use the multilingual and thus more recent data instead of just the translated speeches until 2012. Since for the topic model we do not need perfect grammatically correctly translated texts a simple word-by-word machine translation would have been enough but we could not find a dictionary with all languages that allowed the translation of so many texts. It might also make sense to find some way to incorporate the metadata such as parties and nationality more into the analysis.

References

[1] A. Schofield, M. Magnusson, L. Thompson, and D. Mimno, "Understanding text pre-processing for latent dirichlet allocation," 2017.

[2] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," 1998.

[3] R. Alghamdi and K. Alfalqi, "A survey of topic modeling in text mining," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 1, 2015.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[5] D. Mimno, E. Talley, M. Leenders, H. M. Wallach, and A. Mccallum, "Optimizing semantic coherence in topic models,"

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," 2008.

[7] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” 2002.

[8] K. He, Y. Li, S. Soundarajan, and J. Hopcraft, “Hidden community detection in social networks,” 2017.

[9] S. Pandhre, M. Gupta, and V. N. Balasubramanian, “Community-based outlier detection for edge-attributed graphs,” 2017.