Contents

1 Introduction

With increasing amounts of data being accessible to the public, there has been an ever stronger need for tools which also make the information in this data accessible. The data size has by far exceeded the scope which can be handled by humans. This has given rise to new trends, summed under buzz words like "Big Data" in information technology. The overall goal is to filter what is important and thereby turn unmanageable data sizes into smaller sets of information that can be analyzed manually. Computational linguistic tools play a decisive role in bridging the gap which has opened up between big data and its users.
The digitalization of what used to be available only in print has contributed to the increasing size of databases. This thesis will explore and evaluate the use of clustering and measures from information retrieval to partition parliamentary debates and newspaper articles into topic-based clusters. Moreover, we will examine the fitness of chronology within clusters for novelty detection.
Using a metric approach with vector representations of the documents, topics shall be extracted and, by the help of clustering, be sorted into different topic groups. In each group, the earliest document will be chosen as an indicator for when the topic of this specific cluster arose for the first time. We will differentiate between general topics which are part of daily political discussions and those which are triggered by sudden events and are very prominent during a limited time frame. Term and document frequencies will be combined in such a way that general topics should be neglected in favor of temporarily prominent topics. The goal is to thereby detect topic changes and topics which were not noticed by the public during the time.
Most interesting were the following findings which could be inferred from the measurements:

- Clustering on nouns and a mix of nouns, adjectives and cardinal numbers works best for political debates in terms of within-cluster similarity while cardinal numbers and unknown words yield the highest results for clustering newspaper articles.
- Dimensionality reduction on the document vectors leads to little information loss and breaks down runtime to a fraction of the time needed for clustering the original vectors.
- Topic retrieval based on verbs, adjectives or cardinals alone is difficult but can be very fruitful if enhanced by nouns.

How exactly these results were obtained will be explained in the following chapters. Starting from an overview of the most relevant research on metric approaches to document clustering in Section 2, we will move on to our datasets and how the previously mentioned approach has been implemented to structure this data in Section 3. Finally, the results will be evaluated and the methodology reviewed in Section 4. A critical investigation of the advantages and downsides of a metric data analysis will point out possibilities to refine the methodology as suggested in Section 5 about future work.

2 Approaches to Topic and Topic Change Detection

There are numerous possibilities to capture the change in topics of political texts as debates and newspaper articles. The linguistic levels alone already range from tokens, sequences of adjacent tokens such as bigrams or trigrams (two or three neighboring words), to lemmas and other, more complex relational structures like dependencies between words. A very ad hoc way of searching for topics would be to create a filter which detects specific word combinations, e.g. noun phrases or noun phrases preceded by certain verbs. In the context of parliamentary debates, these could be expressions like Example 1 where the noun phrase following the verb would be extracted as the topic. Another pattern could be Example 2. However, pattern matching is a very restrictive and limited approach as it excludes all topics which are not mentioned in a context defined by these patterns and which will therefore be missed.

1. debattieren über etw.
   'to debate about s.th.'
2. heutige(r) Tagespunkt(e):
   'current topic(s) of debate:'

Newspaper articles have the advantage of usually including an implicit summary of their contents, namely their headlines. As Dorr et al.[3] show with their approach to generate headlines from the first sentences of a document, it is possible to extract the headline from the content but the opposite – i.e. to reconstruct the full content from the headline – is not necessarily true. For instance, headline Example 3 introduces an article the first sentences of which talk about the U.S. ambassador Holbrooke traveling to Afghanistan and how the taliban exchange their members. The two sentences are quite unrelated if no further context is provided, and it is hard to imagine how a headline could be generated from them if the article is actually about general strategy changes.
Headlines are also limited in that they represent subjective descriptions by the article's author, are therefore selective and will not be able to cover all topics treated in the article. Furthermore, headlines tend to be short and are meant to first of all draw the readers' attention to the article rather than informing them about the article's content. Their primary purpose is to raise questions about the article, not to provide answers. Take again the headline in Example 3. All that is mentioned is a change of strategies. Which strategies this is referring to, which parties are involved and what the actual topic of the article is, however, remains unclear. The reader starts asking questions which will not be answered until later in the article's body. Stylistic devices such as alliterations have a similar effect. Examples 4 and 5 sound very catchy but do not tell much about the articles' content. Irony as in Example 6 will be even more unclear in terms of content description, in fact rather raising more questions about what exactly is meant by such expressions than giving information about the article. A positive example for a heading which does provide a topic treated in the article would be Example 7.

3. Strategiewechsel auf beiden Seiten[1]
   'Strategy Change on Both Sides'
4. Rasender Rücktritt[2]
   'Rapid Resignation'
5. Totaler Taliban-Terror[3]
   'Total Taliban Terrorism'
6. FDP-18-Popeye[4]
   'FDP-18-Popeye'
7. Israel rückt wegen Hamas zusammen[5]
   'Israel Closing the Ranks Against Hamas'

What makes headlines impossible to use in this context is the fact that none of the debates has a heading, and comparability between the two corpora would decrease significantly if results were based on very different kinds of data from the corpora. The same holds for strict patterns which are likely to apply to one corpus more than to the other since the vocabulary will differ to some extent between parliamentary debates and journalistic texts.
A more flexible way of finding both general and specific topics should try to capture the overall topicality of the texts and compare the documents according to these characteristics. In a metric approach, such characteristics are frequency counts. How these can be used to find the most central topics in a text document will be explained in the following section.

2.1 Bag-of-Words and the Vector Space Model

The Bag-of-Words approach assumes that the position of the words in the dataset does not matter and only the occurrence of the word irrespective of its context plays a role in differentiating one text from the other [8]. Different kinds of values can be used to "fill" the bag of words. Semantically focused words often come up several times if they do occur [4]. For this reason, term and document frequency have become commonly used measures in bag-of-words. They rely on counts of terms in a document and the number of documents in which the term occurs. A term which is frequent in one document but also frequent in other documents will not contribute to finding interesting patterns in the document collection and will not help in pointing out differences between documents. On the other hand, a term which is frequent in one document but is rare in other documents can be a good indicator for modeling the specifics of the text in which the term occurs. That is why document frequency can serve as indicator of informativeness [4]. As a

[1] 20090212.conll, article no. 26.
[2] Ibid., article no. 33.
[3] Ibid., article no. 32.
[4] Ibid. article no. 22.
[5] Ibid. article no. 20.

consequence, it could be helpful to exclude stopwords or the n most frequent words in the text collection. However, the effect of very frequent and thus in our context meaningless terms can be attenuated by combining the term frequency (tf) with the inverse of the term's document frequency (idf), making a stopword filter obsolete [5]. The idf value is high for rare terms and low for overall very frequent terms. This has the advantage of being adaptive to domain-specific corpora. For example, the *PolMine* data contains a lot of political jargon which will be used very often and therefore does not provide information on the different topics treated in the debates. Unlike inverse document frequency, stopword lists are general and would have to be changed and extended for texts about very specific topics.

$$idf = log \frac{|D|}{df(t)} \tag{1}$$

with D = the set of documents
and *df(t)* = the term's document frequency

$$tf\text{-}idf(t,d) = tf(t,d) \times idf(t) \tag{2}$$

The tf and idf values are calculated for each term of a document, the tf is multiplied by the logarithmically scaled idf and the result is saved in a document vector. Every document will be represented as one vector and put together as a matrix $M \in \mathbb{R}^{|D| \times |V|}$. The idf is logarithmically scaled so that a term which occurs in one document gets *full* weight whereas a term that occurs in all documents gets *zero* weight [4]. Eventually, each document will be represented by one vector, and the text collection will be represented by a set of vectors which will be put into a vector space. This is what is known as the vector space model [5]. It is used for classifying, clustering or retrieving texts with certain properties, e.g. texts containing specific words or word sequences. That can be done by comparing the document vectors to each other or to the query submitted by a user, returning those documents which have similar vectors. The threshold for when documents are similar enough has to be determined empirically by comparing outputs for different similarity values and deciding in favor of the value which delivers the most accurate results.

Cosine similarity and euclidean distance are common measures for comparing vectors based on their proximity or distance, respectively. The higher the cosine similarity between two vectors the more similar are they. Euclidean distance, on the other hand, measures similarity by reducing the distance between vectors. In this case, the smaller the distance between the vector coordinates the more similar are the two vectors.

Euclidean distance is highly dependent on the length of the vectors. To obtain comparable results, it becomes necessary to normalize the vectors to unit length before calculating their euclidean distance. Cosine similarity, by contrast, focuses on the direction of two

vectors measured by the angle or, more precisely, the cosine between the vectors:

$$sim_{cos} = \cos(\theta) = \frac{p \cdot q}{\|p\|\|q\|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}} \tag{3}$$

$$d_{euc} = d(p,q) = \sqrt{(q_1 - p_1)^2 + ... + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{4}$$

When vectors differ with respect to their sizes, the cosine similarity can be calculated without any further preprocessing. For euclidean distance, they have to be normalized by their length or ℓ^2 norm.

$$norm_{l^2} = \|x\| = \sqrt{\sum_{i=1}^{n} x_n^2} \tag{5}$$

Vectors are longer if they have higher values in each dimension. Word frequency vectors will be longer if they contain more frequent words. As a consequence, shorter texts will have shorter vectors since the overall word frequencies are lower. Giving more weight to longer texts, however, would not serve the purpose of finding texts which are similar in content. A short newspaper article about the U.S. real-estate bubble will be just as much related to the economic crisis as a debate on European financial support of Greece. It therefore depends on the application whether cosine similarity or euclidean distance is the most suitable similarity measure. An example where euclidean distance would be preferred over cosine similarity could be clustering cities based on the average travel time between the cities. In this case, the travel times are important, and normalizing them would very much skew the results.

2.2 Dimensionality Reduction: Random Projection Hashing

Term-document matrices tend to become very sparse and have high dimensionality since the overall vocabulary is large but only a small subset occurs in a single document. Linear algebra operations are less efficient on sparse than on dense matrices which raises the question whether the same information could be encoded in more dense vectors. Dimensionality reduction can transform large sparse to small dense vectors. Random projection hashing as introduced by Indyk and Motwani [11] and expanded by Gionis et al. [2] is such a dimensionality-reduction technique. It uses a random matrix $R \in \mathbb{R}^{|V| \times b}$ and computes a dense binary vector d^b for a document vector $d \in \mathbb{R}^{1 \times |V|}$ as follows:

$$d^b = bit(dR) \tag{6}$$

where *bit* is the element-wise function:

$$bit(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The intuition behind random projection hashing is to project points from a high-dimensional vector space into a subspace of less dimensions in such a way that the distances between the points are mostly preserved [15]. For computing the distance between binary vectors, the commonly used measure is the hamming distance which is defined as:

$$d_{hamm}(p,q) = \sum_{i=1}^{n} [y_{p,i} \neq y_{q,i}] \tag{8}$$

The distance between two binary vectors is then equal to the number of bits differing between the vectors. For some purposes, it will be more convenient to deal with a similarity rather than a distance measure. To that end, a hamming distance can be turned into a hamming similarity by calculating $1 - d_{hamm}$.

2.3 K-Means Clustering

The detection of topic changes can be done by grouping all debates into clusters, assuming that each cluster represents one topic. K-Means clustering can help finding such topic clusters. Generally speaking, K-Means is an algorithm to sort vectors in a vector space into clusters. The number of clusters is determined by k $\in \mathbb{N}$, the sorting criterion is the similarity or distance of the vectors to each other. As an instance of a class is assigned to a single and not multiple clusters, K-Means is a kind of hard rather than soft clustering. It is also flat and not hierarchical because the relation between the clusters is undetermined [4]. Picturing the hierarchy as a tree, one element in a determined clustering relation stands for the cluster containing all the objects of its descendants [4].
One downside of hard clustering is that a document cannot belong to more than one cluster. The algorithm has to choose one cluster to which it assigns a document [4]. Taking into account that in this context a cluster is meant to stand for one topic, this may not hold very satisfactory results. Parliamentary debates and newspaper articles are not restricted to one topic per debate or article even though they focus on one or at least few selected topics. This is the reason why the topic of a cluster should be considered a more general concept which sums up the topics in the cluster but itself does not have to be a topic in one or more of the cluster documents.
To obtain the document clusters, random centroid vectors are initialized. As long as a predefined stopping criterion has not been met, all clusters start out as empty sets, the document vectors are assigned to their closest centroids, the vectors are added to their respective cluster and the centroids are recomputed.
The first question to keep in mind is how to pick "good" initial cluster centroids. Generating absolutely random centroids can lead to some clusters staying empty because the centroids are too far away from the data points while most of the documents will be assigned to a few closer centroids. To avoid this, the initial centroids are picked randomly from the set of existing document vectors.
After the centroids are initialized, the similarity between the document vectors and the centroids can be computed and the most similar centroid will be chosen for each

Algorithm 1 K-means ($\vec{\mu_k}$ is centroid of cluster ω_k)

KMeans($\{\vec{x}_1, , ..., \vec{x}_N\}, K$)
$(\vec{s}_1, \vec{s}_2, ..., \vec{s}_k) \leftarrow$ selectRandomSeeds($\{\vec{x}_1, , ..., \vec{x}_N\}, K$)
for $k \leftarrow 1$ **to** K **do**
  $\vec{\mu_k} \leftarrow \vec{s}_k$
end for
while stopping criterion has not been met **do**
  for $k \leftarrow 1$ **to** K **do**
    $\omega_k \leftarrow \{\}$
  end for
  for $n \leftarrow 1$ **to** N **do**
    $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
    $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
  end for
  for k $\leftarrow 1$ **to** K **do**
    $\vec{\mu}_k \leftarrow \frac{1}{\omega_k} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
  end for
end while
return $\{\vec{\mu}_1, ..., \vec{\mu}_k\}$

document. Document clustering aims at creating such a set of clusters that the similarity within the clusters is maximized while between-cluster similarity is minimized [7].
After the documents have been assigned to their closest centroids, the first set of clusters is complete. The average of all vectors in a cluster returns the new centroid for this cluster. The documents will be re-assigned to their most similar centroid and the centroids will be re-calculated until

- the centroids do not move much anymore, i.e. the distance between the previous and the new centroid is small enough (how small exactly, would again have to be determined empirically).
- the documents do not change the cluster anymore.
- the summed similarities of the documents to the centroids do not change much anymore.

When *dense binary* vectors are compared to the centroids, their distance is computed by the normalized hamming distance. The hamming distance is preferred over cosine similarity since the latter would incorrectly return 0 when comparing two 0-bits although this should receive the highest similarity score of 1. The centroid cannot be updated in the same way as for real-valued vectors, either. Averaging over all cluster documents in order to compute the updated centroid would result in a real-valued centroid. In an alternative approach suggested by Geva and de Vries [12], the updated centroid's bits are the medians of all the document vectors in the cluster. If a respective bit is set to 1

in more than half of the document vectors in the cluster, it will also be set to 1 in the centroid.
The final clusters of real-valued or binary vectors can, for example, be validated by repeating the clustering several times and picking the clustering with the lowest residual sum of squares, i.e. the sum of all squared distances between the document vectors and their respective closest centroid.

2.4 Cluster Visualization

What happens after the documents have been clustered? One way of displaying the clusters is to reconstruct the original words from each document in a cluster. These terms can either be given as a complete list or as a selection of terms which are especially informative for a specific cluster. Terms are informative if they provide insights into the overall topic of the cluster. To filter such terms, the number of cluster documents containing the term is calculated and divided by the total number of documents in the cluster. Depending on how strict the filter is applied, only words which are shared between all cluster documents will be shown, or a lower ratio will be chosen.[6] This approach has been implemented in what will be called the `PartiallySharedTerms` method from here on. In order to keep the output size small and manageable, the method displays at most 10 terms per document.

3 Evaluation Setup

3.1 The Corpora

The data serving as the empirical basis for topic extraction and detection of topic changes was taken from *PolMine*[7] out of which a collection of 984 German parliamentary debates from February 1996 until September 2013 was chosen. In a preprocessing step, the debates were tokenized by the OpenNLP tokenizer and tagged by the Citar tagger, both trained on the Tüba-D/Z. Each file contains a single debate which is then split into several debate sections by different speakers of different parties. The association of a debate to its date is central for later analyses of the results.
The *Berliner Tageszeitung*, in the following only called by its nickname *taz*, is a left-wing daily newspaper published in the German capital. It provides the second dataset which is a collection of newspapers from the years 1986 to 2009 and in total comprises 6894 publications. The corpus consists of compressed CoNLL files each of which contains tokens along with their lemmas and more linguistic metadata, e.g. the article ID which helps splitting the file content into articles. The corpus was tokenized by an in-house rule-based tokenizer specifically developed to tokenize the *taz* data. Tagging was done by the Citar tagger, and the SepVerb lemmatizer [6] was used for lemmatization.
Having access to *taz* only in CoNLL format was one reason to also choose CoNLL for *PolMine* as soon as it became available. This makes the analyses of the two datasets

[6] For further explanations about the different ratios, cf. Section 3.2.6.
[7] For further information, see the project website [1].

more comparable because equal filter settings can be configured, such as exclusively picking nouns or participles. Tags as well as lemmas would not have been included in the XML version of *PolMine*.
To give a more concrete idea of the size of the data we are dealing with here, let us look at the number of sections, sentences and words in the *PolMine* and *taz* corpora. One file corresponds to one *PolMine* debate including several speaker turns or to one *taz* newspaper with a number of articles.

	PolMine	**taz**
files	984	6,894
sections	215,081	1,217,682
sections > 10 sentences	73,382	647,413
sections per file	219	177
sentences	4,929,631	22,847,124
words	73,610,885	393,374,363

Table 1: Overall counts by corpus.

PolMine comprises almost 74mio words and 5mio sentences, *taz* approaches 394mio words and 23mio sentences. Having such different corpus sizes at hand adds an interesting dimension to the comparison of cluster configurations even between corpus sizes. Section counts for sections longer than 10 sentences have been included in Table 1 for both corpora but the filter has only been used for *taz* to exclude short articles.

PolMine	*mean*	*median*	*std dev*	*mode*
by files	5009.706	4055.5	3225.574	2162
by sections	22.918	4.0	58.707	1
taz				
by files	3318.391	3331.0	1101.676	3165
by sections	18.663	12.0	22.085	5
by sections > 10 sentences	30.220	24.0	449.5753	11

Table 2: Sentence counts by corpus subdivisions.

It can be seen in Table 2 that *PolMine* debates are considerably longer than *taz* newspapers with respect to the number of sentences, by on average 1700 sentences. Both corpora include a number of outlier files and sections with a high number of sentences. These shift the average number of sentences up while median and mode stay low. This tendency is even stronger in *PolMine* than in the *taz*. This can be inferred both from the higher standard deviation from the mean for *PolMine* file and section divisions and from the fact that the mean and median of *taz* per file are closer to each other than the

mean and median of the *PolMine* texts.
The most frequent sentence number is comparatively low for *PolMine* debates in contrast to *taz* newspapers. Yet, *taz* articles are also quite short, with a most frequent sentence number of 5 and a median of 12. At the same time, their variation from the mean is high which means there are some articles which are very long, considering that the average sentence number is around 18 but the standard deviation from the mean is 22.

PolMine	*mean*	*median*	*std dev*	*mode*
by files	74807.810	59671	45690.900	35390
by sections	365.204	69	629.259	10
taz				
by files	57134.980	57063	18447.240	54778
by sections	321.318	191	399.168	52
by sections > 10 sentences	536.385	430	449.5753	381

Table 3: Word counts by corpus subdivisions.

Similar tendencies as for the sentence counts can be observed for the word counts in Table 3 with respect to the discrepancy between average and median. As is also true for the sentence counts, the most frequent number of words per file and section is much higher for *taz* than for *PolMine* where there are big gaps between mode, median and the overall average. One reason is that many speaker turns are single-sentence contributions to a debate while on the other hand some debate sections are extremely long and possibly closer to scripted than to spontaneous speech. Such a variance is less likely for newspaper texts which are constrained by the page size and number of pages in a newspaper. These characteristics can help to explain why certain clustering configurations work better for one than for the other corpus.

	mean	*median*	*variance*	*std dev*	*mode*
PolMine	15.472	15.749	4.222	2.055	15.294
taz	17.340	17.190	1.893	1.376	16.052

Table 4: Average sentence lengths by corpus. *Taz* sentences are slightly longer than *PolMine* sentences while the latter also vary more. An explanation can be the contrast between (spoken) debates and (written) newspaper texts.

In a token-based approach, the average sentence lengths will be more relevant than when using lemmas. If the average sentence length is relatively short, many words at the beginning of sentences will be capitalized irrespective of their word class. As the corpora are in German and case is one means to distinguish word classes, the tokens have not been converted to lower case. With moderate average sentence lengths of 15.472 for *PolMine* and 17.34 for *taz*, close medians and modes, the effect of capitalized sentence-initial tokens will be minor. This will be confirmed by the experiments conducted on both

tokens *and* lemmas. Which other parameters are varied for the clustering configurations will be explained in the next section.

3.2 Clustering Configurations

Category	**Configuration**
Content division	File
	Section
Tag set	Nominal
	Verbal
	Adjectival
	Other
	Mixed
Word level	Lemma
	Token
Cluster size	4
	10
Vector format	Real-valued
	Binary
Ratio of `Partially`	0.3
`SharedTerms`	0.5
	0.7
	1.0

Table 5: Configuration overview with 256 possible combinations.

3.2.1 Content Division

Both the *PolMine* and the *taz* content can be divided according to files or sections. One file corresponds to one debate or one newspaper, respectively. A section, on the other hand, can be regarded as one speaker turn in a debate or one article in a newspaper. Since 984 debates can be split into more than 215,000 debate sections and 6894 newspapers comprise around 1.2mio articles, a decision has been made in favor of looking at entire files only for *PolMine* to speed up both runtime and time in analyzing the data which increases with every dimension taken into consideration.
The same could *not* be done for *taz* as the structure of a newspaper is very different from that of a parliamentary debate. Newspapers usually combine topics from politics, economics, and culture, they range from regional to international affairs, from opinionated editorials to factual reports. Each newspaper contains mixes of topics, making the documents quite similar and uniform. Clustering such documents would not yield very informative results in terms of distinctive, unique topics. For this reason, the *taz* has

been split according to articles rather than newspapers, and a filter has been introduced to ignore all short articles with less than 10 sentences. Articles shorter than 10 sentences were assumed to be either comments or other material which would slow down clustering without contributing much to the variety of topics.

3.2.2 Tag Categories

Both corpora provide tokens with their respective part-of-speech tag. A tag filter which only allows words with a certain tag can influence the quality and content of the clustering output very much. Some word classes are already excluded by the stopword list, such as personal pronouns or particles like "nicht" in Example 8.

8. Ich mag das *nicht*.
   'I do*n't* like that.'

9. Ich glaube, *dass* es morgen warm wird.
   'I think *that* it'll be warm tomorrow.'

Corpus-independent stopwords are removed before checking the tags. Although low tf-idf values should take care of such words not being included in the calculations, stopwords like conjunctive "dass" as in Example 9 still does appear in the `PartiallySharedTerms` output. As this makes result interpretations unnecessarily difficult, and processing all stopwords takes up a lot of runtime while not contributing to a fruitful output, each token is checked against a list of predefined stopwords. All tokens which occur in this list will be ignored.
Nominal tags are NN for common nouns, NE for proper nouns and TRUNC for truncated words which are often nominal compounds separated by a hyphen. Nouns are expected to be good indicators of topics in a text. TRUNC, however, turned out to be surprisingly specific in the topics which it returns. Among these are, for example, political descriptions as "CDU-" and "parlaments-", country or regional descriptions like "US-" and "Gaza-", or more sector-related terms such as "Wirtschafts-", "Börsen-" and "Chemie-". While phrases related to politics such as names of political parties may in the *PolMine* corpus be more likely to play the role of stopwords, i.e. they occur in all or most of the texts and throughout a variety of different topics, economic jargon and country or region indicators can help to learn more about the topic of the debate or newspaper.

10. CDU-, parlaments-, US-, Gaza-, Wirtschafts-, Börsen-, Chemie-
    'CDU, parliamentary, US, Gaza, economic, stock market, chemical' (adjectival)

Using verb tags has both major advantages and disadvantages. They make it easier to differentiate between positive and negative pieces of text since many verbs are inherently positive or negative. Some examples for negative verbs which occurred quite frequently are "klagt", "droht", "abgesagt" or "vermisst". At the same time, verbs can be quite ambiguous. Would "belastet" refer to bad relations or polluted air? Which topic would a debate or newspaper be about where "steigt" is mentioned? It could equally be about raised prices or about asset values at the stock market, one negative, the other positive.

Tag category	**Tag**	**Description**	**Example**
Nominal	NN	Common noun	[an der] Spitze '[at the] top'
	NE	Proper noun	Kreml-Entourage 'Kreml entourage'
	TRUNC	Composite component	Staats- [und Regierungschef] '[chief] of state [and chief executive]'
Verbal	VVFIN	Finite verb	gelingt 'succeeds'
(full verbs)	VVINF	Infinitive	bekräftigen 'support'
	VVIZU	Verb with "zu"	voranzutreiben '[in order to] promote'
	VVPP	Perfect participle	geführt 'led'
Adjectival	ADJA	Attributive adjective	[eine] offene [Gesellschaft] '[an] open [society]'
	ADJD	Adverbial/predicative adjective	[eine Hand] voll [Menschen] '[a] handfull of [people]'
Other	CARD	Cardinal number	1998 cf. above
	FM	Foreign word	Business Crime Control cf. above
	XY	Unknown word	[Klassen] A [bis] D '[classes] A [to] D'
Mixed	NN NE TRUNC	⟩	cf. *Nominal*
	ADJA ADJD	⟩	cf. *Adjectival*
	CARD		cf. *Other*

Table 6: Selected tags from the Stuttgart-Tübingen-TagSet (STTS [10]) with examples from the *PolMine* debate BT_13_086.xml.

11. klagt, droht, abgesagt, vermisst, belastet, steigt
    'moans/complains/sues, threatens, canceled, misses, stresses, rises'

Adjectives can be very useful in their function of giving further information about places and people. Such terms are given in Example 12.

12. iranisch, hamburgisch, international, links
    'Iranian, "Hamburgian", international, left-wing'

Other word classes are cardinal numbers including years, foreign language material and unknown words. Year specifications may hint at the year in which the debate was held or in which the newspaper was published. Foreign language material and unknown words are rather uncommon and could therefore be especially useful in making out the difference between one debate or newspaper and another.
After experimenting with the tag filter set, a mixed category was introduced. It includes all nominal tags (NN, NE, TRUNC), all adjective tags (ADJA, ADJD) and cardinal numbers (CARD). This decision was based on the informativeness of the produced output. A great proportion of those nouns and adjectives returned by `PartiallySharedTerms` is related to specific topics in a debate, and many of the numbers are dates close to that of the earliest document in the cluster. Choosing this combination of tags makes interpreting the output much more fruitful.

3.2.3 Word Layers

The CoNLL format in which both corpora have been made available provides text input as both tokens or lemmas, i.e. as word form including inflectional and declination affixes or as base form. Tokens are a more precise representation of the text as inflection and declination indicate closer relations between words. Base forms, on the other hand, give a better impression of the overall occurrence of a word without taking into consideration different forms of the same base. Using the base form as processing unit also helps reducing the dimensions of the frequency vectors by adding further vector elements only for new words, not for every new word form. For example, a vector using tokens would have three dimensions to represent the distinct word forms "beschließt", "beschlossen" and "beschloss" whereas a vector using lemmas would only have one dimension in which all three words would be represented as "beschließen".

13. beschließen, beschließt, beschlossen, beschloss
    'to decide, decides, decided (3^{rd} pers. pl./ past part.), decided (3^{rd} pers. sg.)'

3.2.4 Vector Formats

Dimensionality reduction by means of random projection hashing leads to two alternative vector representations: the original sparse tf-idf document vectors and the binary vectors of reduced dimensionality. For real-valued vectors, the cosine similarity is used to cluster the documents into topic-related clusters. Binary vectors rely on the hamming distance for document clustering. It is thus not only the vector format but also the K-Means

algorithm which varies depending on the kind of vector input that is passed to it. Real-valued vectors are retrieved directly from the tf-idf matrix whereas the binary vectors are the result of transforming the tf-idf vectors into binary vectors via random projection hashing.
Conventionally, the number of bits, which corresponds to the number of randomly generated vectors and the length of the dimensionality-reduced vectors, is a power of 2. The range of possible values is arbitrarily large, and one has to be picked. This can be done by comparing the cosine similarity between one document and a set of documents to the corresponding hamming distance. The original high-dimensional space is euclidean space, and cosine similarity is used to measure similarity. In the space with fewer dimensions, hamming distance is chosen instead but the relation between the vectors should not change from high- to low-dimensional space. Therefore, the extent to which the cosine similarities in euclidean space correlate to the hamming distances in low-dimensional space is an indicator of how well the dimensionality reduction preserves the original relations between the document vectors. A higher correlation between cosine similarities and hamming distances indicates a closer mapping between the spaces. The correlation will become higher with increasing the number of bits. At some point, however, the reduction of computation costs outweighs the gained precision of the results.

Number of bits	128	256	512	1,024	2,048	4,096
Correlation	0.953	0.975	0.986	0.992	0.995	0.996
Hashing time (in ms)	22,207	44,805	90,905	180,256	361,895	729,226

Table 7: Correlation between cosine similarities and hamming distances for specific numbers of bits, retrieved by comparing all ***PolMine*** debates with their 10 most similar debates, using lemmas from the mixed tag category. A number of bits of 1024 is considered the best compromise between a good-enough correlation of the similarity measures and runtime.

Table 7 shows how for *PolMine* the correlation between cosine similarities and hamming distances increases with higher numbers of bits. Again, one has to keep in mind that runtime doubles when the number of bits used to represent each document is doubled. Therefore, the best compromise between efficiency and precision is a bit number of 1024. For more bits, the correlation goes up by less than 0.005, and a correlation of 0.992 is close enough to the optimum of 1.0 to choose 1024 as the standard number of bits for binary vectors in the following measurements.

3.2.5 Cluster Sizes

Finding the best average cluster size is a trade-off between cluster coherence and large enough clusters. If the clusters become too small, they will not cover general but very specific topics. At the same time, clusters should not become too large as the range of topics will be too diverse to still consider a cluster as summary of one topic. An average cluster size of 10 has been chosen as the larger cluster size, 4 will be the average of a

smaller cluster size setup. Similarity scores are expected to increase for clusters with on average 4 documents per cluster.

3.2.6 `PartiallySharedTerms` Ratio

The quality of the clusters will be evaluated based on the words in a cluster. For a word to be representative for a cluster, the tf-idf value has to be relatively high compared to the other words in the document, and the word has to occur in several documents from the cluster. The ratio for the `PartiallySharedTerms` method defines in how many documents from the cluster a word has to occur. A ratio of 0.5 will make sure that all words have to be present in at least half of the documents in the cluster, 1.0 would mean only words are displayed which come up in *all* documents from the cluster. A low ratio of 0.3 ensures that also for heterogeneous clusters with little word overlap among documents the list of topic words is not empty. All ratios have been included in the following analysis.

4 Results and Evaluation

The question of how to evaluate the results already arose at the beginning stage of the project. The data is unlabeled which in this case means that no predefined topics are given for a specific parliamentary debate or *taz* article. Since the goal was to also detect topics that are less prominent and less known to the public, manually labeling a subset of the data and thereby excluding all other topics from the results would not have helped in achieving the overall goal. Therefore, a quantitative comparison of expected and actual topic labels was not possible.
An alternative qualitative analysis investigates the `PartiallySharedTerms` output more closely. As the number of clusters exceeds an amount which can be handled in total, some clusters have to be selected for more fine-grained analyses. This can be done in two ways: a predefined list of topics serves as selection criterion and one cluster per topic is picked, or a set of clusters is randomly chosen. These techniques can also be used in combination to enhance one another by providing multiple perspectives on the same dataset. The earliest document in the cluster will play an important role in that it locates the cluster on the time scale. A topic may come up at different times, spread even across centuries. What we are interested in here is the first occurrence of a topic. Another sign of quality of a cluster can be if the earliest topic occurrence dated by the cluster is equal or close to the actual occurrence of a topic. The latter has to be determined by looking at reliable resources which provide dates of historic events.
While the `PartiallySharedTerms` output cannot be captured by means of numeric results which are directly comparable, this is possible for the cosine and hamming similarity measures.[8] Due to its reasonable size, the first assessment of the different clustering configurations was done on the *PolMine* corpus. Tables 8 and 9 contain

[8] The hamming distances have been converted to hamming similarities as described in Section 2.2 so that for both real-valued and binary clustering higher scores indicate better results than lower scores.

all numeric results. The `PartiallySharedTerms` output will only come into play in the evaluation where the quantitative analysis of the within-cluster cosine and hamming similarity will be enhanced by a qualitative evaluation of the topics returned by `PartiallySharedTerms`.

4.1 Examining Configuration Results

∅ Cosine similarity	Median	Standard deviation	Variance	Tag category	∅ Cluster size	Word layer
0.675	0.665	0.184	0.034	Mixed	4	Lemma
0.670	0.647	0.184	0.034	Nominal	4	Lemma
0.661	0.636	0.198	0.039	Nominal	4	Token
0.658	0.592	0.227	0.051	Mixed	4	Token
0.640	0.629	0.168	0.028	Adjectival	4	Lemma
0.637	0.588	0.243	0.059	Verbal	4	Lemma
0.636	0.574	0.241	0.058	Verbal	4	Token
0.618	0.607	0.175	0.031	Adjectival	4	Token
0.597	0.592	0.227	0.051	Other	4	Token
0.597	0.585	0.226	0.051	Other	4	Lemma
0.543	0.502	0.211	0.045	Nominal	10	Token
0.543	0.496	0.211	0.045	Nominal	10	Lemma
0.542	0.497	0.208	0.043	Mixed	10	Lemma
0.540	0.495	0.207	0.043	Mixed	10	Token
0.539	0.464	0.240	0.058	Verbal	10	Lemma
0.534	0.451	0.232	0.054	Verbal	10	Token
0.521	0.509	0.164	0.027	Adjectival	10	Lemma
0.496	0.465	0.165	0.027	Adjectival	10	Token
0.486	0.467	0.180	0.032	Other	10	Lemma
0.478	0.458	0.192	0.037	Other	10	Token

Table 8: Clustering results for real-valued vectors of ***PolMine***, sorted by cosine similarity. Across cluster sizes and word layers, [Mixed] and [Nominal] outperform the other categories.

The cosine similarities for real-valued clustering on *PolMine* display moderate up to good results. With a maximum cosine similarity of 0.675 for the setting [Mixed; 4;

Lemma][9] and a minimum of 0.478 for [Other; 10; Token], the average cosine similarities on real-valued vectors cover a data range of approx. 0.2. Individual scores almost exhaust the entire bandwidth between 0 and 1. With a median deviating up to 0.08 from the mean for [Verbal; 10; Token], this gives further indication about how some single cosine similarity scores pull the average slightly towards their score. A low number of outliers can be inferred from the standard deviation and variance measures: Standard deviation goes from 0.164 up to 0.243, variance from 0.027 to 0.059, meaning that the majority of the cosine similarity scores deviates at most by 0.243 from the mean.

∅ Hamming similarity	**Median**	**Standard deviation**	**Variance**	**Tag category**	**∅ Cluster size**	**Word layer**
0.769	0.743	0.111	0.012	Verbal	4	Lemma
0.769	0.744	0.102	0.010	Nominal	4	Lemma
0.768	0.750	0.101	0.010	Nominal	4	Token
0.765	0.748	0.098	0.010	Mixed	4	Lemma
0.764	0.749	0.104	0.011	Verbal	4	Token
0.764	0.742	0.102	0.010	Mixed	4	Token
0.752	0.745	0.084	0.007	Adjectival	4	Lemma
0.743	0.737	0.079	0.006	Adjectival	4	Token
0.743	0.733	0.092	0.008	Other	4	Token
0.742	0.735	0.098	0.010	Other	4	Lemma
0.713	0.674	0.103	0.011	Verbal	10	Lemma
0.712	0.685	0.092	0.008	Nominal	10	Token
0.711	0.684	0.096	0.009	Nominal	10	Lemma
0.708	0.682	0.091	0.008	Mixed	10	Lemma
0.708	0.684	0.090	0.008	Mixed	10	Token
0.707	0.675	0.095	0.009	Verbal	10	Token
0.691	0.682	0.060	0.004	Adjectival	10	Lemma
0.684	0.673	0.071	0.005	Other	10	Lemma
0.681	0.675	0.061	0.004	Adjectival	10	Token
0.681	0.673	0.065	0.004	Other	10	Token

Table 9: Clustering results for binary vectors of ***PolMine***, sorted by hamming similarity. As a major contrast to the results from real-valued clustering, the best configuration is [Verbal; Lemma].

[9] In the following, reference to the clustering configurations will be made in the format [tag category; cluster size; word layer].

As expected, smaller clusters obtain higher cosine similarity. The reason for this has already been hinted at in Section 3.2.5. Small clusters contain few documents, large clusters contain many documents. For a large cluster to achieve the same similarity score, more documents have to be similar to each other whereas in a smaller cluster only few documents have to be similar.
The distinction between lemmas and tokens does not seem to play such a big role. For all except for the [Nominal; 10; Token/Lemma] configuration, lemmas return slightly better results than the corresponding token configurations. Yet, the difference in cosine similarity stays within at most 0.022 in the case of [Adjectival; 4; Token] vs. [Adjectival; 4; Lemma].
Due to the different ways in which the cosine and hamming similarity are calculated, their results cannot be compared directly. Although the absolute hamming similarity averages for binary document vectors of *PolMine* are higher than the cosine similarities of real-valued vectors and the low standard deviation of the hamming similarities implies that the single hamming similarities are less spread out than those of the cosine similarities, the same value in cosine and hamming similarity still stands for different kinds of similarities. It is only due to the project setup with clustering on tf-idf valued vectors that the cosine similarities do not exhaust the full bandwidth of values between -1 and 1 but stay within the same bounds as the hamming distances. For vectors with positive and negative values, this approach will not be valid anymore.
The hamming similarity scores by themselves are quite convincing. A reason can be that the noise which the real-valued vectors contain is removed by random projection hashing. Noise can be any information provided by the input data which does not contribute to solving the document clustering task. Examples are high-frequency and function words as well as topic-unrelated terms. After the hashing, only those features – here words and their frequencies – are left which help in distinguishing the documents from each other. The more distinctive the features are, the easier will it be to cluster similar documents into topic-specific clusters.
The configuration [Verbal; 4; Lemma] yields the best hamming similarity of 0.769. This is surprising both because the [Verbal] category does not receive high scores for the real-valued vectors and because the same configuration with tokens for binary vectors ends up behind nominals and some of the [Mixed] category results. It first seems like the lemmatization has more of an effect on verbs than on other categories. When looking at the absolute hamming similarities, though, it becomes clear that it is rather the proximity of the values to each other than the difference of lemmatized and tokenized input which causes the results to be ordered in such a way.
As the *taz* articles are on average shorter than the *PolMine* debates, it is not surprising that the similarities in the *taz* clusters are higher. The fewer words a section contains, the more words would overlap with a similar section. For longer texts, there would still be a relatively small overlap even if they are similar. The high cosine similarity medians of 0.891 for [Other; 4; Lemma] and 0.814 for [Other; 10; Token] indicate that the single inside-cluster similarities are even higher than the averaged cosine similarities of 0.801 and 0.747 respectively, and some outliers slightly decrease the average similarity score.

∅ Cosine similarity	Median	Standard deviation	Variance	Tag category	∅ Cluster size	Word layer
0.801	0.891	0.272	0.074	Other	4	Lemma
0.747	0.814	0.272	0.074	Other	10	Token
0.652	0.620	0.204	0.042	Verbal	4	Lemma
0.637	0.612	0.224	0.050	Adjectival	4	Lemma
0.515	0.480	0.174	0.030	Verbal	10	Token
0.469	0.431	0.190	0.036	Adjectival	10	Token

Table 10: Clustering results for real-valued vectors of ***taz***, sorted by cosine similarity. The [Other] category which does worst for *PolMine* yields best cosine similarities for *taz*, even irrespective of the cluster size.

∅ Hamming similarity	Median	Standard deviation	Variance	Tag category	∅ Cluster size	Word layer
0.887	0.889	0.104	0.011	Other	4	Lemma
0.885	0.884	0.105	0.011	Other	4	Token
0.855	0.840	0.113	0.013	Other	10	Lemma
0.852	0.834	0.113	0.013	Other	10	Token
0.762	0.731	0.110	0.012	Verbal	4	Lemma
0.757	0.731	0.113	0.013	Adjectival	4	Lemma
0.748	0.715	0.113	0.013	Verbal	4	Token
0.737	0.713	0.108	0.012	Adjectival	4	Token
0.709	0.690	0.077	0.006	Verbal	10	Lemma
0.703	0.687	0.080	0.006	Adjectival	10	Lemma
0.692	0.674	0.075	0.006	Verbal	10	Token
0.679	0.664	0.069	0.005	Adjectival	10	Token

Table 11: Clustering results for binary vectors of ***taz***, sorted by hamming similarity. Hamming distances for *taz* display the same ordering as the cosine similarities.

Unlike *PolMine* where [Mixed] scores best for cosine and [Verbal] best for hamming similarity, the *taz* results display similar tendencies for real-valued and binary vectors. The ranks of the various tag categories are exactly the same: [Other] ranges before [Verbal] and [Adjectival]. In this respect, *taz* results are more stable across the two vector formats than *PolMine*.
Another difference to *PolMine* is that for *taz* the cluster size has less of an effect on the similarity scores than the tag category. All [Other] configurations receive the highest

cosine or hamming similarity, irrespective of whether the cluster size is 4 or 10. While all results of *PolMine* are completely split into the upper half for clusters of on average 4 members and the lower half for clusters of on average 10 documents, the two cluster sizes are more mixed in the *taz* result tables. One reason may be that the *taz* corpus is much larger than the *PolMine* set of debates and it is more likely that there will be many similar articles than there are similar debates. A second explanation could be that some configurations with an average cluster size of 10 do so well on *taz* that they outperform other configurations with smaller clusters.

Tag category	**Cluster size**	**Word layer**	**t(RVC)**	**t(BC)**	**t(RVC)/ t(BC)**
Mixed	10	Token	143197	112	1 278.545
Mixed	4	Lemma	181529	146	1 243.349
Mixed	4	Token	226993	186	1 220.392
Nominal	4	Token	153589	171	898.181
Mixed	10	Lemma	128232	147	872.327
Nominal	10	Token	86071	106	811.991
Nominal	10	Lemma	75325	95	792.895
Nominal	4	Lemma	132053	173	763.312
Adjectival	4	Token	44675	196	227.934
Adjectival	10	Token	26627	128	208.023
Adjectival	4	Lemma	29297	157	186.605
Adjectival	10	Lemma	18014	107	168.355
Verb	4	Token	25281	171	147.842
Verb	10	Lemma	10367	94	110.287
Verb	4	Lemma	19460	191	101.885
Verb	10	Token	12751	129	98.845
Other	10	Token	5566	80	69.575
Other	4	Token	8755	153	57.222
Other	10	Lemma	4385	84	52.202
Other	4	Lemma	7774	163	47.693

Table 12: Clustering runtimes, t in ms on an Intel Xeon E5-2698 v4 server CPU with 2.2GHz, for ***PolMine***, sorted by the proportion of clustering with binary vectors (BC) being faster than clustering with real-valued vector input (RVC).

Tag category	Vocabulary size	
	Token	*Lemma*
Mixed	596,885	485,713
Nominal	432,992	366,348
Adjectival	96,652	60,611
Other	33,979	33,937
Verbal	31,319	27,144

Table 13: Sorted vocabulary sizes of ***PolMine*** for all tag categories and word layers.

The positive conclusion about using binary rather than real-valued vectors drawn from the high hamming similarity scores can be strengthened by looking at the runtime comparison of the two setups in Table 12. For all configurations, the clustering on binary vectors is faster than on real-valued vectors. In the worst case, binary clustering is 48 times, in the best case more than 1200 times faster than the real-valued vector clustering. This difference becomes even more important when working on larger datasets.
For *taz*, better runtimes made it possible to obtain binary clustering results for all cluster sizes and word layers with the tag categories [Adjectival], [Other] and [Verbal]. For the slower clustering on real-valued vectors, only the most and least successful configurations of [4; Lemma] and [10; Token] could be run and evaluated in a reasonable amount of time. Given the computing resources and time available, results for [Nominal] and [Mixed] which have the largest vocabularies could not be obtained. This underlines the importance of dimensionality reduction for clustering of serious datasets.

4.2 Retrieving Topics

The next two sections will focus on how well the chronology of topic clusters fits an expected chronology of the topics, and on the process of novelty detection in *PolMine* and *taz*. Compared to the analysis and interpretation of the cosine and hamming similarity results, the evaluation of the topics returned by the `PartiallySharedTerms` method is subject to quite a few more restrictions. These will become clear in the following exploration of the results.

4.2.1 Fitness of Chronology

As the [Mixed; 4; Lemma] configuration had the best result for cosine similarity in *PolMine* it will be the starting point for a deeper analysis of the clustering results. If not indicated, the examples will be taken from the output of `PartiallySharedTerms` for this configuration. This specific setting includes words from three tag categories, namely nominals and adjectives along with cardinal numbers. It thereby allows better insights into which tag categories help most in detecting known and later also unknown or less known topics.

Topic **'Description'**	**Key word**	**Date**
BSE, Rinderwahn(sinn) 'BSE'	BSE	1996
Der große Lauschangriff 'wide eavesdropping attack'	Lauschangriff	1998
Mehrwertsteuererhöhung 'increasing VAT'	Mehrwertsteuer(erhöhung)	1998
Massenarbeitslosigkeit 'mass unemployment'	Massenarbeitslosigkeit	1998
KFOR-Einsatz im Kosovo 'KFOR intervention in Kosovo'	KFOR, Kosovo	1999
Expo Hannover 'world exhibition in Hannover'	Expo	2000
Hartz IV 'unemployment benefit reform'	Hartz(-Konzept)	2004
AKW-Laufzeitverlängerung 'lifetime extension of npps'	Laufzeit(verlängerung)	2009
Fukushima 'Fukushima nuclear catastrophe'	Fukushima	2011
Energiewende 'energy transformation'	Energiewende	2011
Stuttgart 21 'Stuttgart 21 development project'	Stuttgart	2011

Table 14: Predetermined topics with keywords to search for in the `PartiallySharedTerms` output, and the expected earliest occurrence of the topic.

A downside of using tf-idfs on the word level is that only single-word expressions such as the keywords in Table 14 will be matched, phrases as in the topics column either cannot be found at all or have to be searched for in parts, as e.g. "Kosovo" – instead of "Kosovo conflict" – which may then refer to another topic about Kosovo. The willingness of German nouns to combine to compounds makes things a bit easier, especially since longer compounds condense a lot of information in a single term, as for example "Mehrwertsteuererhöhung" does. A complex topic or an event can thereby be described in just one word with losing only parts of the information that would be present in a longer description. Those cases in which a topic keyword is found only once in the entire set of clusters are quite rare. In most cases, the keyword comes up several times in different clusters. However, it does not always refer to the topic of interest, as has been shown with the "Kosovo" vs. "Kosovo conflict" example. In order to determine how closely a topic is

mapped to a date, it is therefore necessary to not only look at the keywords themselves but also at other words in a cluster and their semantic proximity to the keyword. The more similar words are found in a cluster, the more likely the cluster and thus the keyword is indeed about the topic under inspection. If the keyword is from a very different semantic field than the rest of the partially shared terms, the cluster is most likely not about the keyword topic.

Topic keyword(s) and expected 1^{st} occurrence		**Cluster dates (RVC)**	**Cluster dates (BC)**
BSE	1996	–none–	1998, 2001*
Lauschangriff	1998	1998*, 2005, 2006*	2006
Mehrwertsteuererhöhung	1998	1999, 2006(4)	1998, 2000, 2005, 2006(5)
Massenarbeitslosigkeit	1998	1996(2)	1996(2)
KFOR	1999	2003*, 2012	2000
Kosovo	1999	1998(2), 1999(4), 2000(3), 2001(2), 2003, 2012	1998, 1999(2), 2000, 2004(2)
Expo	2000	–none–	2000(2)
Hartz(-Konzept)	2004	2002(3), 2003(3), 2004, 2005, 2006	2002, 2006
Laufzeit(verlängerung)	2009	2002, 2005, 2006	2000, 2006(2), 2010(2)
Fukushima	2011	2010, 2012*	–none–
Stuttgart	2011	2003, 2010(2), 2011(2)	2010*, 2011
Energiewende	2011	2010, 2011(3), 2012(3)	2005, 2006, 2011, 2012(2)

Table 15: The expected 1^{st} occurrences of the predefined topics with their actual occurrence dates in the clusters retrieved from real-valued and binary vectors of the [Mixed; 4; Lemma] configuration for ***PolMine***. Clustering on binary vectors yields slightly better results in terms of correct date–topic assignment.

* : one cluster with noticeably more occurrences of the term than the other clusters

(n): number of clusters about the specified topic with the same date

To get an overall impression of the keyword search, Table 15 lists the dates of all clusters which contained the respective keyword for both the real-valued (RVC) and binary clustering (BC) results. While the latter guesses the date right twice for the topics "Mehrwertsteuererhöhung" and "Expo", the former only has one date which agrees with the expected first occurrence, for the topic "Lauschangriff". If the correct date is not among the cluster dates, binary clustering results are closer to the expected first occurrence more often than real-valued clustering output, namely three times for the topics "BSE", "KFOR" and "Laufzeit(verlängerung)" compared to only once for "Fukushima". However, there are no clusters with "BSE" in the real-valued clustering

output and therefore no date to compare the binary results to. The same is true for "Fukushima" which did not occur at all in the partially shared terms from binary clustering. This makes "KFOR" and "Laufzeit(verlängerung)" the two topics to consider, and for both, binary clustering does better.
In general, the clusters tend to be dated rather too early than too late compared to the expected earliest occurrence of the topic. Both clustering on real-valued and binary vectors return the same early date for the topics "Massenarbeitslosigkeit" and "Kosovo": for the former, the expected earliest date would be 1998, the clusters suggest 1996; for the latter, the expected earliest date is 1999 but the suggestion is 1998. Other topics are also dated too early without thereafter providing the correct date. In terms of clusters, this means that for those topics there are one or more clusters with an early date, in some cases clusters with later dates, but no cluster with the correct date. That tendency of early dates can be explained by the way the dates are retrieved from the clusters. The single dates of the documents in a cluster can differ quite a bit. From these dates, only the earliest date is chosen to locate the cluster on the timeline. Yet, the date of the document in which the topic keyword was found could still be later in time than the earliest document in the cluster. All that the cluster date can do is provide an *earliest* date. It can*not* give a *latest* date, a defined time frame or an order in which the debates were held or the articles were published.
Another reason for topics being dated too early could be the difficulty in determining an exact date for the expected first occurrence of a topic. "Lauschangriff", for example, was an event which can be dated quite accurately whereas "Mehrwertsteuererhöhung" keeps coming up in political debates and newspapers from time to time. Unique events are easier to date than long-term processes or re-occurring events. Looking at the suggested dates in Table 15, this impression is partially confirmed: "BSE", "Lauschangriff" or "Expo" come up in rather few clusters – in at most three – compared to "Mehrwertsteuererhöhung" – in six clusters from real-valued clustering, in eight clusters from binary clustering. Then again, more dates would be expected for "KFOR" which is a peace-keeping force entered in the year 1999 and which is still active today. A topic may be discussed in parliament without any concrete actions being taken. The expected first occurrence is that moment when a decision is made about a debated issue. The discussion which precedes the decision will not be captured by the expected earliest date but could be captured by the suggested date in a cluster. To verify the assumption that sudden events are dated more accurately and are found in less clusters than long-term processes or re-occurring events, more data and a more fine-grained selection of topic clusters would be necessary. Such a selection would involve an evaluation of which document from the cluster does actually treat the topic of the cluster and which one only mentions the topic keyword.
Certain words in a cluster indicate that a general keyword refers to a specific topic, e.g. if "Stuttgart" and "Infrastrukturprojekt" occur in the same document it is likely about "Stuttgart 21". Without such terms, it is sometimes difficult to know what a topic refers to, e.g. when "Stuttgart" is found in the same document as "Bürgerdialog" the question is whether "Bürgerdialog" really refers to "Stuttgart 21". Example 14 lists all words which supports (or may support) the argument of a cluster being about "Stuttgart 21".

It has to be noted that these words are *not* part of the *same* cluster but have been collected from several clusters in all of which "Stuttgart" is mentioned.

14. Bahn, Bahnhof, Bürgerdialog, Demonstration, Infrastrukturprojekt, Polizei, Protest, Volksentscheid, Volksabstimmung, 21
    'German Rail, train station, civic dialogue, demonstration, infrastructure project, police, protest, plebiscite, popular vote, 21'

In a qualitative analysis, it is a subjective judgment which terms are considered belonging to the same topic. For example, are "Hartz (IV)" and "Kündigungsschutz" related closely enough to claim that a document with both terms treats "Hartz IV" as a topic? The current approach has been liberal in that the co-occurrence of two possibly related terms was enough to indicate that they are about the same topic.

15. Kündigungsschutz
    'security of tenure'

On the other hand, there are terms which are never used directly but which are described by other terms in the cluster, e.g. "Weltwirtschaftskrise" is not among the top 10 of shared terms in any cluster but "Bank", "Finanzmarkt", "Währung", "Rettungsschirm" – all of these in the same cluster – as well as "Bankensektor", "Bankenaufsicht" – all of the latter in another cluster — do occur. While this cannot be investigated further here, this may be an interesting observation for future projects about topic detection based on a list of subtopics.

16. Weltwirtschaftskrise; Bank, Bankenaufsicht, Bankensektor, Finanzmarkt Rettungsschirm, Währung
    'world economic crisis; bank, bank supervision, banking industry, financial market, bailout fund, currency'

Some notes shall be added about the `PartiallySharedTerms` output for other tag categories besides [Mixed]. It turns out that some of those work well for clustering documents, at least when looking at their similarity scores for *taz*, but cannot be used for finding topics. [Verbal] and [Other] outputs are hardly interpretable in terms of lexical analyses of the most frequent words shared between cluster documents. [Other] contains mainly numbers without any further context and is only helpful in combination with words from other categories, as has been exemplified with "Stuttgart" – "21".
The same is true for verbs, irrespective of how specific they are. Consider Example 17 from binary clustering of *taz* [Verbal; 4; Lemma], where the first half of the verbs is quite general and the second half rather specific. Both kinds of verbs do not hint at a concrete topic unless they are enhanced by nominals or adjectives which clarify the context. Example 18 lists the context words of the topic search for "Afghanistan" in the previously analyzed *PolMine* configuration and gives an idea of which categories could go well together with verbs to provide more information about the topic of a document or cluster.

17. geben, lassen, sagen; evakuieren, vergiften, protestieren
    'to give, to let, to say; to evacuate, to poison, to protest'

18. Afghanistan, arabisch, Israel, israelisch, Kosovo; Aktionsplan, Ausschuss, Außenminister, Bundeswehr, Gefahrenabwehr, militärisch, Schutzgebiet, Sicherheitsrat, Taskforce, Waffenstillstand, Zuwanderung
    'Afghanistan, Arabic, Israel, Israeli, Kosovo; plan of action, commission, minister of foreign affairs, army, active defense, military, protectorate, security council, task force, ceasefire, immigration'

Adjectives are mostly either very general, such as "seltsam" and "neu", or extremely specific, like "liebeslaubengroß" or "kunstbetriebsintern" from the configuration [Adjectival; 10; Lemma] for clustering binary vectors of *taz*. Both extremes do not provide any insights into the topic of the debate or article. Therefore, adjectives can only be interpreted as belonging to a specific topic if more adjectives or words from other categories, especially nouns, narrow down the scope.

19. seltsam, neu; liebeslaubengroß, kunstbetriebsintern
    'strange, new; as big as a lovers' bower, art scene internal'

All these observations about the [Other], [Verbal] and [Adjectival] categories confirm that a [Mixed] tag category can be especially useful for analyzing the terms which are partially shared between documents in a cluster. The combination of different tag categories narrows the scope of otherwise general words like numbers, verbs or adjectives.

4.2.2 Novelty Detection

The detection of novel topics turned out to have its small and larger pitfalls. Rather than providing a list of novel topics, a "critique of maneuver" will be the focus of this section. Examples of *possibly* novel or unknown topics will be given to illustrate if not problems then peculiarities of the two datasets.
It is not so much the lack of new topics but rather the decision of what exactly could qualify as a novelty which makes the task of novelty detection more difficult than the previous evaluation of the fitness of chronology. In addition, political jargon includes words which are very different from what people outside of politics would use even though they are equally aware of the subject. An extreme case are law terms as in Example 20, both found in the [Mixed; 4; Lemma] *PolMine* clusters from real-valued vectors.

20. Berufsausbildungssicherungsgesetz, Berufsvormündervergütungsgesetz
    'vocational training act, professional guardian compensation act'

German compounding makes such one-word constructions easily possible while they are not so easy to understand for laymen. Even if these topics had been under public discussion one would hardly remember the exact terms. It is also common for complex concepts to be referred to by an alternative nickname in public. Such nicknames would be more likely to be encountered in newspapers than in more formal political debates.
Irrespective of the inherent complexity of some terms in question, the proportion of laws

and law acts among the unknown topics is observed to be relatively large. This is first of all due to the context in which the parliamentary debates of *PolMine* took place, namely on the level of the national government which discusses more general topics than the state governments. Therefore, it is not surprising that the members of parliament spend a great part of their speeches on passing law acts or trying to prevent them from passing. More law terms can be found in Example 21.[10]

21. Altautoverordnung, Arbeitsfrderungsgesetz, Zuwanderungsgesetz
'old vehicles act, work promotion act, immigration act'

The lack of new topics in the `PartiallySharedTerms` output could also be caused by the filter method applied to the vocabulary in each document. Only those terms will be ranked high up by the `PartiallySharedTerms` method which have a high tf-idf value. Completely new topics – unless being the central topic of a debate or an article – may have a very low term frequency and could only have a high tf-idf value if their document frequency is also low. However, such terms will be outdone by other words which have a low document frequency but at the same time have a higher term frequency in the document(s) in which they occur. Low tf-idf values will lead to the new topics not showing in the `PartiallySharedTerms` output as the top 10 shared terms are selected by their tf-idf value. It is therefore possible that the tf-idf values are not high enough for new, unknown topics to be listed in the `PartiallySharedTerms` output.

5 Future Work

5.1 Dimensionality Reduction via Auto-Encoders

For this thesis, random projection hashing has been used to reduce the dimensions of the document vectors. An auto-encoder is a neural network which can fulfill the same task and could be used for a more advanced approach to clustering German parliamentary debates and newspapers. Auto-encoders aim at representing the input data in a compressed format which for document vectors would result in reducing the dimensions of the vectors. Whether the compression has actually been successful can be checked by uncompressing the data again and comparing the output to the original input. The more similar the output is to the input, the more did the encoder manage to learn the most important information or characteristic features from the input.
Encoder, code layer and decoder build the core of the network[9]: The encoder produces the compressed representation of the input the result of which can be seen at the code layer. The decoder reconstructs the input from its compressed representation on the code layer. Comparing the input to the output can be a good means to evaluate the quality of the encoder. For documents, this would mean to compare the topics of the input to the topics of the output document.

[10] As the [Adjectival], [Verbal] and [Other] categories have turned out to be unsuited for this kind of analysis and shared terms from *taz* are only available from these categories, no *taz* examples have been added at this point.

5.2 Considering Semantic Similarity

A downside of using tf-idf matrices for clustering documents is that they do not take the similarity of words into account. In such matrices, terms as "Politiker", "Politikerin" and "Staatsmann" are considered to be just as different as "Politiker" and "Verhandlung". Similar words are more likely to co-occur in a document. However, if in one article or speech "Politiker" and "Verhandlung" occur while another writer or speaker uses "Staatsmann" and "Beratung", these texts are treated as unrelated.

22. Politiker, Politikerin, Staatsmann; Verhandlung, Beratung
    'politician (male, female), statesman; negotiation, briefing'

For document retrieval and clustering alike, results could therefore be improved by using representations that take the similarity of words into account. Documents with similar topics but different wordings would thereby become more similar than they are in the current tf-idf matrix approach. Possible techniques to include such semantic information are word embeddings [14] or WordNets.
Word embeddings can also help in finding words which are similar to a keyword or search term. As shown with the search term "Weltwirtschaftskrise" and the context terms in Example 16 in Section 4.2.1, a specific keyword may not be found in a corpus but the topic it denotes could possibly be detected by semantically similar search terms.
Another advantage of word embeddings is that alternative spellings and misspelled words would become equal to their canonical version in a word embedding. In a tf-idf matrix, a misspelled word is just as dissimilar from the correct word as a completely different word would be. Word embeddings could get rid of this imbalance.

5.3 Extension Beyond Unigrams

The tf-idf matrix is based on counts of unigrams which can be either tokens or lemmas. For non-compositional terms, though, it can be more informative to take context into consideration as well. Looking at the phrases from Example 23, the words "Weg" and "bringen" are just as general and uninformative in isolation as "Farbe" and "bekennen". The contrary is true for the four-gram "auf den Weg bringen" and the bigram "Farbe bekennen" which have a very specific meaning.

23. etw. auf den Weg bringen; Farbe bekennen
    'drive forth s.th.; to show one's colors'

These more informative n-grams can be filtered from all n-grams using measures such as pointwise mutual information (PMI) and specific correlation, the generalization of PMI for more than two variables [13].

6 Applications

A database with the clustered *PolMine* debates could be used to relate a debate to *taz* articles which treat the same topic as the debate. With one of the debates at hand, a set

of relevant *taz* articles should be selected and suggested to the users. The suggestions would be based on the similarity between the returned *taz* articles and the debate. The quality of the output could be checked by comparing the debate topic to the topic of the article(s): the more closely related the topics are, the higher is the quality of the document clustering mechanism.
Another application could be to offer to the users a list of topics which are treated in the debates and/or articles. By clicking on a topic, a set of documents about this topic would be presented to the users. They could define the number of documents to retrieve. The higher the number, the more will the documents' content deviate from the topic. An alternative would be to return only those documents which are very much and not only somewhat related to the topic of interest. This would guarantee a higher quality of the output under the risk of leaving the users with fewer documents than they were hoping to find.
The database can also be a tool for any person interested in politics to learn more about a certain topic with a collection of preselected, topic-related material at hand. With the inverted indices and the list of centroids being accessible once the clustering has been done, users could even look at the document clusters in relation to each other. Such inter-cluster relations can be hierarchical or relational in the sense that clusters with similar centroids will treat similar topics. This can be exploited to generate intermediate topics to group more specific topics from clusters together into subsets with broader topics. Debates and articles from various levels of specificity could then be provided to the users.

7 Discussion and Conclusion

Parliamentary debates and newspaper articles have been taken as two examples of political text genres to retrieve topics by clustering based on real-valued and binary vectors. Results have been calculated for different clustering configurations, making use of five tag categories and two word layers to filter the input texts, and of two cluster sizes for the average cluster size of the returned set of clusters.
The distinction between lemmas and tokens does not make much of a difference with respect to the similarity scores. What is more important is the tag category filter applied to the texts. The tag categories with best similarity scores differ considerably between *PolMine* and *taz*: [Mixed] and [Nominal] are best for *PolMine* and [Other] performs worst while [Other] is better than all tested categories for *taz*. This implies that the tag category filter may have to be adapted to the kind of input data.
Inferring cluster topics from the `PartiallySharedTerms` output is easiest for the [Mixed] tag category whereas [Adjectival], [Other] and [Verbal] by themselves are not topic-specific enough. Enhancing them with words from the [Nominal] tag category and, for [Adjectival], having more adjectives which restrict the context of topic-unrelated adjectives can ease topic retrieval. Context terms and similar words also help in determining the exact topic of a cluster if the keyword itself is ambiguous – which it most often is. A comparison between suggested and expected earliest occurrence of a topic as

well as the analysis of the `PartiallySharedTerms` output could be made more precise by excluding clusters in which the topic keyword occurs but the other terms indicate that the cluster has a different main topic.
Novelty detection requires a framework to measure what constitutes a "novel" topic. Without such a framework, it becomes difficult to separate commonly known from less known and unknown topics. The depth of analysis necessary for novelty detection as attempted here turned out to be outside of the scope of this thesis.
All the more strong is the argument that dimensionality reduction speeds up runtime for binary clustering without resulting in much information loss. On the contrary, the `PartiallySharedTerms` output is slightly more accurate in assigning dates to topics than the clustering on real-valued vectors. At the same time, clustering of binary vectors can be up to 1200 times faster than for real-valued vectors which will be even more significant for larger datasets. Finally, the high correlation between the cosine and hamming similarity scores shows that random projection hashing succeeds in reducing dimensionality while preserving the relations between the document vectors.
Space for improvement leaves the general random projection hashing used here. Hashing could be made more optimal by taking into account that all tf-idfs will be zero or positive and, as a consequence, all vector components are positive. The random vectors which are used for creating the binary document vectors are generated for the whole vector space half of which will not be covered by the tf-idf vectors. In a refined version, random vectors would be generated only in that part of the vector space in which the document vectors lie. This would ensure that the set of document vectors is split by the random vectors and that it can never happen that all document vectors are located on the same side of a random vector. It would be worth following up on this topic to further improve the binary clustering approach now that it has been shown that binary vectors are successful in maintaining the main characteristics of the real-valued vectors while significantly speeding up clustering.

References

[1] Andreas Blätte: *Purpose of the PolMine-Project*, `http://polmine.sowi.uni-due.de/polmine/` (accessed January 18, 2017).

[2] Aristides Gionis, Piotr Indyk, and Rajeev Motwani (1999). "Similarity Search in High Dimensions via Hashing", in *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, pp. 518–529.

[3] Bonnie Dorr, David Zajic, and Richard Schwartz (2003). "Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation", in *Proceedings of the HLT-NAACL 2003 Workshop on Text Summarization*, pp. 1–8.

[4] Christopher D. Manning and Hinrich Schütze (2003). *Foundations of Statistical Natural Language Processing*. Cambridge, London: MIT Press.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Cambridge, London: University Press.

[6] Daniël de Kok (2014). "TBa-D/W: a large dependency treebank for German", in *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13), Tbingen, Germany*.

[7] Daniel Jurafsky and James H. Martin (1999). *Speech and Language Processing*. New Jersey: Prentice Hall.

[8] Daniel Jurafsky and James H. Martin (2016): *Speech and Language Processing* (draft). New Jersey: Prentice Hall.

[9] Geoffrey E. Hinton and Ruslan R. Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks", in *Science* (313:5786), pp.504–507.

[10] Institut für Maschinelle Sprachverarbeitung. *STTS Tag Table (1995/1999)*, `http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-table.html` (accessed May 06, 2017)

[11] Piotr Indyk and Rajeev Motwani (1998). "Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality", in *Proceedings of the 30^{th} Symposium on Theory of Computing*, pp. 604–613.

[12] Shlomo Geva and Christopher M. de Vries (2011). "TOPSIG: Topology Preserving Document Signatures", in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 333-338.

[13] Tim Van de Cruys (2011). "Two Multivariate Generalizations of Pointwise Mutual Information", in *Proceedings of the Workshop on Distributional Semantics and Compositionality (DiSCo'2011)*, pp.16–20.

[14] Tomas Mikolov et al. (2013). "Distributed Representations of Words and Phrases and their Compositionality", in *Advances in Neural Information Processing Systems 26*, pp. 3111–3119.

[15] William B. Johnson and Joram Lindenstrauss (1984). "Extensions of Lipschitz Mappings into a Hilbert Space", in *Contemporary Mathematics* (26:1), pp.189–206.