Kritische Studien zur Demokratie

Gregor Wiedemann

Text Mining for Qualitative Data Analysis in the Social Sciences

A Study on Democratic Discourse in Germany

Springer VS

3. Integrating Text Mining Applications for Complex Analysis

The last chapter already has demonstrated that Text Mining (TM) applications can be a valid approach to social science research questions and that existing studies employ single TM procedures to investigate larger text collections. However, to benefit most effectively from the use of TM *and* to be able to develop complex research designs meeting requirements of established QDA methodologies, one needs specific adaptions of several procedures as well as a systematic integration of them. Therefore, this chapter introduces an integrated application of various TM methods to answer a specific political science research question. Due to the rather abstract character of the research question, customarily it would be a subject to manual qualitative, interpretive analysis on a small sample of documents. Consequently, it would aim for extensive description of the structures found in the data, while neglecting quantitative aspects. One meta-objective of this study is to show that also TM methods can contribute to such qualitative research interests and, moreover, that they offer opportunities for quantification. To guide the analysis for the research question on *democratic demarcation* briefly introduced in Section 1.3, I propose a workflow of three complementary tasks:

1. document retrieval to identify (potentially) relevant articles from a large corpus of newspaper data (Section 3.1),

2. (unsupervised) corpus exploration to support identification and development of categories for further analysis (Section 3.2),

G. Wiedemann, *Text Mining for Qualitative Data Analysis in the Social Sciences*,
Kritische Studien zur Demokratie, DOI 10.1007/978-3-658-15309-0_3

3. classification of context units into content analytic categories for trend analysis, hypothesis testing and further information extraction (Section 3.3).

Each task of this workflow is described by its motivation for a QDA scenario, its specific implementation or adaptation, its optimal application with respect to requirements of the example study, and approaches for evaluation to assure quality of the overall process.

3.1. Document Retrieval

3.1.1. Requirements

When exploring large corpora, analysts are confronted with the problem of selecting relevant documents for qualitative investigation and further quantitative analysis. The newspaper corpus under investigation $\mathcal{D}$ comprises of several hundreds of thousands of articles (see Section 1.3). The absolute majority of them might be considered as irrelevant for the research question posed. Thus, the first of the three tasks introduced in this analysis workflow is concerned with the objective to reduce a large data set to a smaller, manageable set of potentially relevant documents. This can be related clearly to an *ad hoc* task of IR comparable to search applications such as library systems or web search engines:

> "The ad hoc task investigates the performance of systems that search a static set of documents using new topics. This task is similar to how a researcher might use a library—the collection is known but the questions likely to be asked are not known" (Voorhees and Harman, 2000, p. 2).

Nonetheless, IR for QDA differs in some respects from standard applications of this technology. In standard scenario of ad hoc IR, users generally have a specific, well defined information need around specific topics or concrete (named) entities. This information need can be described with a small set of concrete key terms for querying a collection. Furthermore, the information need can be satisfied with a

relatively small number of documents to be retrieved. Search engine users rarely have a look on more than the first page of a retrieval result, usually displaying the ten most relevant items matching a query (Baeza-Yates and Ribeiro-Neto, 2011, p. 267). Thus, most retrieval systems are optimized with regard to precision[1] among the top ranks of a result while recall[2] might be neglected.

In contrast to this standard scenario of IR, I identify different requirements when applying it for large scale QDA concerned with rather abstract research questions:

- Research interests in QDA often cannot be described by small keyword queries.[3] How to formulate a reasonable query for *documents containing expressions of democratic demarcation*? The information need of my example study rather is contained in motifs, language regularities and discourse formations spread over multiple topics which require an adapted approach of IR.

- While standard IR focuses on precision, an adapted QDA procedure has to focus on recall as well. The objective of this task is the reduction of the entire collection of a newspaper to a set of documents which contains most of the documents relevant to the research question while keeping the share of documents not related to it comparatively small.

- Related to this, we also need to know, how many documents from the entire collection should be selected for further investigations.

To meet these special requirements of QDA, this section proposes a procedure of IR using *contextualized dictionaries*. In this approach, a

[1] Precision is considering the share of actual relevant documents among all documents retrieved by an IR system.

[2] Recall expresses the share of relevant documents retrieved by an IR system among all relevant documents in the searchable collection.

[3] Doubtlessly, there are examples for QDA information needs which work well with simple keyword queries. For example, an analysis of political debates on the introduction of a legal minimum wage in Germany certainly can query an indexed corpus for the term *Mindestlohn* and try to filter out not domestically related retrieval results afterwards.

query is not based on single terms compiled by the content analyst. Instead, the query is automatically built from a set $\mathcal{V}$ of reference documents. Compared to the problem of determining concrete key terms for a query, it is rather easy for analysts to manually compile a collection of 'paradigmatic' documents which reflect topics or language use matching their research objective. Retrieval for a set of documents $\mathcal{D}' \subseteq \mathcal{D}$ with such a reference collection $\mathcal{V}$ is then performed in three steps:

1. Extract a substantial set of *key terms* from the reference collection $\mathcal{V}$, called dictionary. Terms in the dictionary are ranked by weight to reflect difference in importance for describing an analysis objective.

2. Extract term *co-occurrence statistics* from the reference collection $\mathcal{V}$ and from an additional generic comparison corpus $\mathcal{W}$ to identify language use specific to the reference collection.

3. *Score relevancy* of each document in the entire global collection $\mathcal{D}$ on the basis of dictionary and co-occurrence statistics to create a ranked list of documents and select a (heuristically retrieved) number of the top ranked documents for $\mathcal{D}'$.

Related Work

Heyer et al. (2011) and Rohrdantz et al. (2010) introduce approaches of interactive exploratory search in large document collections using data-driven methods of pattern identification together with complex visualizations to guide information seekers. Such contemporary approaches to IR also address some of the requirements described above. Nevertheless, in their data-driven manner they allow for identification of interesting anomalies in the data, but are less suited to integrate prior knowledge of social scientists to select document sets specific to a research question. To include prior knowledge, the approach of using documents for query generation is a consequent idea within the VSM of IR, where key term queries are modeled as document vectors

for comparison with documents in the target collection (Salton et al., 1975). The proposed approach extends the standard VSM approach by additionally exploiting aspects of meanings of topic defining terms captured by co-occurrence data. Co-occurrence data has been used in standard IR tasks for term weighting as well as for query expansion with mixed results (van Rijsbergen, 1977; Wong et al., 1985; Peat and Willett, 1991; Holger Billhardt et al., 2000). These applications differ from the approach presented here, as they want to deal with unequal importance of terms in a single query due to term correlations in natural language. The method presented in this chapter does not globally weight semantically dependent query terms by co-occurrence information. Instead, in CA analysts are often interested in certain aspects of meaning of specific terms. Following the distributional semantics hypothesis (see Section 2.2.1), meaning may be captured by contexts better than just by isolated terms. Therefore, relevancy is scored based on similarity of individual contexts of single query terms in sentences of the target documents in $\mathcal{D}$ compared to observed contexts from the reference collection $\mathcal{V}$. In case of my example study, this approach may, for example, not only capture the occurrence of the key term "order" in a document contributing to its relevancy score. In addition, it captures whether the occurrence of the term "order" is accompanied by terms like "liberal", "democratic" or "socialist" which describes contents of interest much more precisely than just the single term.

This section describes the details of this adapted IR approach and is organized as follows: After having clarified the motivation, the next section presents different approaches of dictionary extraction for automatic query generation. The subsequent parts explain how to utilize ranked dictionaries together with co-occurrence data for document retrieval. Finally, an evaluation of the approach is presented.

3.1.2. Key Term Extraction

The generation and usage of dictionaries is an important part of quantitative CA procedures (Krippendorff, 2013). Dictionaries in the

context of CA are basically controlled lists of key terms which are semantically coherent with respect to a defined category (e.g. terms expressing religious beliefs, emotions or music instruments). These lists provide the basis of code books and category systems within CA studies. Usually dictionaries are crafted by analysts in manual processes. Yet, their creation also can be supported by computational methods of key term extraction. For the proposed approach of document retrieval, I utilize automatically extracted dictionaries describing characteristic vocabulary extracted from a reference collection. To exploit dictionaries for document retrieval different methods of key term extraction might be used. Each method puts emphasis on different text statistical aspects of the vocabulary which, of course, leads to different lists of extracted key terms as well as to different levels of semantic coherence between them. Consequently, we can expect varying results for the retrieval process when utilizing such dictionaries. To evaluate which method of key term extraction produces the most valuable result for our IR task, three methods are compared:

- Term Frequency–Inverse Document Frequency (TF-IDF),
- Topic Models, and
- Log-likelihood (LL).

But first of all, I describe how I compiled the reference collection $\mathcal{V}$ for key term extraction to retrieve documents related to the subject of democratic demarcation.

Compiling a Reference Collection about Democratic Demarcation

The objective of compiling a collection of reference documents is to create a knowledge resource to support the process of IR for complex, rather abstract research interests on qualitative data. Not only vocabulary in form of a dictionary is extracted from this collection, but also co-occurrence statistics of terms, which yield a more meaningful description of typical language use within the collection. Thus, the collection should match the research interest of the content analyst

in the best possible way. It should be representative in vocabulary and contextual meaning of terms for the content, which is targeted in the later IR process. Therefore, reference documents should be selected carefully by the analysts in consideration of representing domain knowledge and specific language use of interest. The selection of documents needs to be justified as an important initial step throughout the overall process. Moreover, one has to consider shifts in language use over time as well as between different genres of text. For example, it makes a difference of taking scientific or administrative documents for a reference collection to retrieve newspaper articles, instead of using also newspaper articles. The decision to use documents of a different genre (as done in this example study) may be made consciously to cover influences of language use specific to certain actors from other discourse arenas. For retrieval of documents from a long time period, the reference collection should also contain documents from a similar time frame to capture shifts and developments of language use appropriately.

For my example study on "democratic demarcation", I decided to rely on five editions of the "Verfassungsschutzbericht" as a basis for the reference collection—one of each decade since the first report from 1969/70. "Verfassungsschutzberichte" are official administrative reports of the German domestic intelligence service Bundesamt für Verfassungsschutz (BfV) published by the Bundesministerium des Innern (BMI). They report on developments, actors and topics state officials perceive as threat to the constitutional democratic order of the FRG (Murswiek, 2009). In this respect they are an excellent source to extract language of "democratic demarcation" within the German discourse on internal security and democracy. The compiled reference collection consists of:

- Verfassungsschutzbericht 1969/1970, (BMI 1971)
- Verfassungsschutzbericht 1979, (BMI 1980)
- Verfassungsschutzbericht 1989, (BMI 1990)
- Verfassungsschutzbericht 1998, (BMI 1999)

- Verfassungsschutzbericht 2009, (BMI 2010)

All reports were scanned and OCR-ed. Then, the following preprocessing steps (see Section 2.2.2) were applied: Sentences were separated and tokenized, tokens were lemmatized and transformed to lower case. For IR purpose, I need a reasonable number of reference documents in length comparable to newspaper articles. For this, I split the five rather long documents (50–200 pages per report) into smaller pseudo-documents. Sequences of 30 successive sentences were pooled to pseudo-documents to mimic boundaries of contextual coherence for the term extraction approaches via TF-IDF and topic models. The final reference collection $\mathcal{V}$ consists of 137,845 tokens in 15,569 sentences and 519 pseudo-documents.

TF-IDF

The TF-IDF measure is a popular weighting scheme in IR (Baeza-Yates and Ribeiro-Neto, 2011) to express how informative a single term is to describe specific content of a document, or in our case the whole reference collection $\mathcal{V}$. For this, each term t is weighted by its frequency tf within the entire collection on the one hand, and inverse document frequency on the other hand:

$$w_t = \mathrm{tf}(t, \mathcal{V}) \times \log \frac{|\mathcal{V}|}{\mathrm{df}(t, \mathcal{V})} \tag{3.1}$$

The underlying assumption is that a term t is more important if it is more frequent within the reference collection. At the same time, t is more informative to describe a document if it is present only in few documents, instead of (nearly) every document of the entire reference collection. This is expressed by the inverse of the document frequency $\mathrm{df}(t, \mathcal{V})$.

Topic Models

Statistical topic models infer groups of thematically coherent terms (see Section 2.2.3) which can be used to extract relevant vocabulary

from a collection $\mathcal{V}$ of paradigmatic documents (Wiedemann and Niekler, 2014). Topic models infer probability distributions of terms in topics $\boldsymbol{\beta}$ and topic distributions in documents $\boldsymbol{\theta}$. Topics in the LDA model (Blei et al., 2003) are assumed as a fixed number K of underlying latent semantic structures. Posterior probabilities $P(t|\beta_k)$ for each word t from the entire vocabulary of collection $\mathcal{V}$ can be inferred for any of the topics $k \in (1, \ldots, K)$ by sampling-based inference. Terms with a high probability in the kth topic represent its determining terms and allow for interpretation of the meaning of an underlying thematic coherence. In contrast to TF-IDF, the topic model approach for term extraction can take account of the fact that terms do not occur independently of each other. Thus, highly probable topic terms may be utilized to compile another valuable dictionary of keywords from a collection.

Probability distributions from $\boldsymbol{\beta}$ can easily be transformed into weights to receive a ranked list of terms describing the reference collection $\mathcal{V}$. In the simplest case the weight of a term in the dictionary can be defined as the sum of its probability values within each topic $\sum_{k=1}^{K} P(t|\beta_k)$. In comparison to term frequency counts in a collection, the probability weight of a term in a topic represents its contribution to the topic context. Even if this topic has relatively low evidence in the collection (represented by low values $\theta_{\cdot,k}$) a term can have high probability $P(t|\beta_k)$ within this topic. To not overly bias the ranks in the dictionary with very improbable topics and their words, a normalization strategy is needed. One solution is to additionally use term frequency to weight the terms within the corpus. As the final weight w_t of a term t in the dictionary, I define:

$$w_t = \log(\text{tf}(t, \mathcal{V})) \sum_{k=1}^{K} P(t|\beta_k) \tag{3.2}$$

where K is the number of topics and $\text{tf}(\cdot, \mathcal{V})$ the term frequency within the reference collection $\mathcal{V}$. By using log frequency the effect of high frequency terms is dampened.

Table 3.1.: Word frequency contingency table for term extraction (Rayson and Garside, 2000).

	$\mathcal{W}$	$\mathcal{V}$	Total
Frequency of t	a	b	$a+b$
Frequency of other words	$c-a$	$d-b$	$c+d-a-b$
Total	c	d	$c+d$

In a topic model usually topics with undesired content can be identified. Some topics group syntactic terms, such as stop words or foreign language terms (AlSumait et al., 2009). Other topics, although capturing coherent semantic structure, may be considered as irrelevant context for the research interest. In contrast to other keyword extraction methods which neglect interdependence of terms, the topic model approach allows to exclude such unwanted semantic clusters. Before calculating term weights, one simply has to identify those topics not representing meaningful structures and to remove them from the set of the K term-topic distributions $\beta_{1:K}$. This can be an important step for the analyst to influence the so far unsupervised dictionary creation process and a clear advantage over other methods of key term extraction.

Log-Likelihood

'Keyness' of terms not only can be calculated on basis of the collection $\mathcal{V}$ itself, but with the help of a (generic) comparison corpus $\mathcal{W}$. Occurrences of terms as events are observed in the comparison collection. Based on these observations expectations of term frequencies within the target collection can be calculated. Then, deviations of the actually observed frequencies from the expected frequencies are compared using a statistical test. For language data, the log-likelihood ratio test (Dunning, 1993) has proven to provide useful results. Rayson and Garside (2000) use this approach to calculate Log-likelihood (LL) statistics for each term by the contingency Table 3.1. Expected fre-

quency values E_i in one corpus are calculated on the basis of observed frequencies in the other by the formulas $E_1 = c(a+b)/(c+d)$ and $E_2 = d(a+b)/(c+d)$. The LL statistic allows for conclusion of the significance of relative frequency differences between the two corpora, although thresholds for significance levels might be hard to define (ibid.). Consequently, the LL statistic can be employed as term weight w_t directly and is calculated as follows:

$$w_t = 2(a \log(a/E_1) + b \log(b/E_2)) \tag{3.3}$$

Whether the difference indicates an over- or underuse in the target corpus $\mathcal{V}$ compared to the comparison corpus $\mathcal{W}$ can be derived from the comparison of the relative frequencies of t within both corpora ($a/c < b/d \Rightarrow$ overuse in $\mathcal{V}$). The overused terms can then be sorted by their weights in decreasing order, resulting in a list of characteristic terms specific to the target corpus.

I used `deu_wikipedia_2010_100K-sentences` as comparison corpus $\mathcal{W}$—a corpus of 100,000 sentences randomly chosen from the German Wikipedia provided by the "Leipzig Corpora Collection" (Biemann et al., 2007).[4] To utilize it as comparison corpus, sentences need to be preprocessed exactly the same way as for $\mathcal{V}$, i.e. tokenization, lemmatization and lowercase reduction.

Extracting Terms

From the paradigmatic collection of the "Verfassungsschutzberichte" specific vocabulary is extracted with each of the three methods described above (TF-IDF, Topic Models and Log-Likelihood). Terms can be listed in ranks by sorting their term weights w_t in decreasing order. This results in a list of ranked words which can be cut to a certain length N. For the further process, I decide to take the first

[4] The Leipzig Corpora Collection provides language resources carefully maintained by computational linguists. Its corpora may be seen as representative of common language characteristics not specific to a certain domain or topic. Corpora containing up to one million sentences can be downloaded from http://corpora.uni-leipzig.de.

$N = 750$ terms of each list as a dictionary to build a query q for document retrieval. Cut-outs from the top and the bottom of the three extracted dictionaries are displayed in Table 3.2.

3.1.3. Retrieval with Dictionaries

Dictionaries can be employed as filters in IR systems to reduce general collections to sub-collections containing sets of documents of interest for further analysis. Using a dictionary of ranked terms for IR can be formulated as a standard VSM problem in combination with 'term boosting'. For this, dictionary terms are translated into a query q of unequally weighted key terms. Prior knowledge of unequal importance of terms is incorporated into query processing via factors based on term ranks. A simple VSM-scoring function can be computed for a document $d \in \mathcal{D}$ and a dictionary-based query q as follows:[5]

$$\text{score}_{\text{VSM}}(q, d) = \text{norm}(d) \times \sum_{t \in q} \text{tf}(t, d) \times \text{boost}(t) \qquad (3.4)$$

This baseline formula for querying a collection with a ranked dictionary only considers term frequency *tf*, term weight based on dictionary rank *boost* and a factor for document length normalization *norm*.

[5] Basic VSM scoring for IR is described in (Baeza-Yates and Ribeiro-Neto, 2011, p. 61ff). An example with 'term boosting' is implemented in Apache's famous Lucene index: http://lucene.apache.org/core/2_9_4/api/all/org/apache/lucene/search/Similarity.html.

Table 3.2.: Key terms in the reference collection of "Verfassungsschutz" reports automatically extracted with three different methods.

Rank	TF-IDF	w_t	Topic Model	w_t	Log Likelihood	w_t
1	rechtsextremistisch	282.987	deutschland	1.003	rechtsextremistisch	1580.916
2	partei	228.409	politisch	0.941	partei	1244.438
3	gruppe	203.883	partei	0.842	organisation	1128.310
4	organisation	196.795	organisation	0.686	politisch	1019.042
5	kommunistisch	196.351	deutsch	0.594	deutschland	993.258
6	mitglied	191.183	rechtsextremistisch	0.560	rechtsextremist	718.553
7	deutsch	187.280	mitglied	0.540	kommunistisch	704.539
8	deutschland	182.775	gruppe	0.460	extremistisch	664.406
9	politisch	172.092	person	0.422	terroristisch	614.421
10	extremistisch	168.843	ziel	0.392	linksextremist	596.035
11	rechtsextremist	166.631	aktivität	0.319	aktion	585.447
12	person	160.400	kampf	0.315	bestrebung	558.702
13	bundesrepublik	159.994	kommunistisch	0.295	linksextremistisch	557.763
14	terroristisch	157.004	aktion	0.294	aktivität	530.493
15	linke	156.779	insbesondere	0.286	bundesrepublik	528.302
16	linksextremist	156.195	mehren	0.285	linke	521.425
17	türkei	153.537	september	0.284	gruppe	510.109
18	gewalttat	151.343	bundesrepublik	0.265	gewalttat	493.557
...	...	...	...	...	...	...
746	prinzip	35.839	bonn	0.017	kosovo-albaner	30.112
747	rechtfertigen	35.839	besondere	0.017	partisi	30.112
748	saugen	35.839	ablehnen	0.017	provider	30.112
749	zerstören	35.839	zumindest	0.017	prozeß	30.112
750	zumindest	35.839	beziehen	0.017	strasser	30.112

Usually IR weightings also consider the Inverse Document Frequency (IDF) of a term as a relevant factor to take account of unequal contribution of terms to the expressiveness of a query. Since ranks of the dictionary already represent information about unequal importance, I skip the IDF factor. Instead, rank information from the dictionary needs to be translated into a boosting factor for the scoring function. I suggest a factor ranging between 0 and 1 for each term t

$$\text{boost}(t) = \frac{1}{\sqrt{\text{rank}(t)}} \tag{3.5}$$

which reflects that the most prominent terms in a dictionary of N terms are of high relevancy for the retrieval process while terms located nearer to the end of the list are of lesser, more equal importance.

Document length normalization addresses the problem of identifying relevant documents of all possible lengths. This is necessary because the longer the document, the higher the chance that it contains dictionary terms. Without length normalization, relevancy scores of long documents would outweigh shorter ones even if the latter ones contain a higher share of query terms. I utilize pivoted unique normalization as introduced in Singhal et al. (1996). Pivotal length normalization slightly lowers relevancy scores for shorter documents of a collection $\mathcal{D}$ and consequently lifts the score for documents after a pivotal value determined by the average document length. The normalization factor for each document is computed by:

$$\text{norm}(d) = \frac{1}{\sqrt{(1 - \text{slope}) \times \text{pivot} + \text{slope} \times |U_d|}} \tag{3.6}$$

where U_d represents the set of unique terms occurring in document d and *pivot* is the average number of unique terms over all documents of the collection $\mathcal{D}$, computed by:

$$\text{pivot} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} |U_d| \tag{3.7}$$

When evaluation data is available, the value for *slope* might be optimized for each collection. Lacking a gold standard for our retrieval

task, I set $slope = 0.7$ which has proven to be a reasonable choice for retrieval optimization in various document collections (Singhal et al., 1996, p. 6).

Further, the *tf* factor should reflect on the importance of an individual term relative to the average frequency of unique terms within a document. Average term frequency per document is computed by:

$$\text{avgtf}(d) = \frac{1}{|U_d|} \sum_{t \in U_d} tf(t, d) \tag{3.8}$$

Moreover, log values of (average) term frequencies are used, to reflect on the fact that multiple re-occurrences of query terms in a document contribute less to its relevancy than the first occurrence of the term. Putting it all together, the final scoring formula yields a dictionary-based document ranking for the entire collection:

$$\text{score}_{\text{dict}}(q, d) = \text{norm}(d) \times \sum_{t \in q} \frac{1 + \log(\text{tf(t,d)})}{1 + \log(\text{avgtf}(d))} \times \text{boost}(t) \tag{3.9}$$

3.1.4. Contextualizing Dictionaries

The scoring function $\text{score}_{\text{dict}}$ yields useful results when looking for documents which can be described by a larger set of key terms. When it comes to more abstract research interests, however, which aim to identify certain meanings of terms or specific language use, isolated observation of terms may not be sufficient. Fortunately, the approach described above can be augmented with co-occurrence statistics from the reference collection $\mathcal{V}$ to judge on relevancy of occurrence of a single key term in our target document. This helps not only to disambiguate different actual meanings of a term, but also reflects the specific usage of terms in the reference collection.

Therefore, I compute patterns of co-occurrences (see Section 2.2.3) of the $N = 750$ terms in our dictionary with each other, resulting in an $N \times N$ matrix $\mathbf{C}$, also called Term-Term-Matrix (TTM). Co-occurrences are observed in a *sentence* window. Significance of a co-occurrence is calculated by the *Dice* statistic, a measure to compare

the similarity of two sets, in our case all sentences A containing one term a and all sentences B containing another term b:

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3.10}$$

Using this measure instead of more sophisticated co-occurrence significance tests, such as Log-likelihood, is preferred in this case to achieve comparable value ranges for different corpora. The Dice statistic ranges between 0 and 1, i.e. the cases set that a and b never, or respectively, always occur together in one sentence. Although it is a rather simple metric, the Dice statistic reflects syntagmatic relations of terms in language relatively well (Bordag, 2008). This is useful for dealing with an unwanted effect, I experienced when experimenting with co-occurrence data to improve the retrieval mechanism. Co-occurrences of terms in the sentences of a reference collection may reflect characteristics in language use of the included documents. However, certain co-occurrence patterns may reflect general regularities of language not specific to a collection of a certain domain or topic (e.g. strong correlations between the occurrence of term pairs such as *parents* and *children*, or MWUs like *United States* and *Frankfurt Main* in one sentence). Applying co-occurrence data to IR scoring tends to overemphasize such common language patterns in contrast to meaningful co-occurrence of term usage specific to the reference collection. To mitigate this effect, we can apply a 'filter' to the extracted co-occurrences.

Instead of using the TTM $\mathbf{C}$ solely based on the reference collection $\mathcal{V}$, I filter the co-occurrence data by subtracting a second TTM $\mathbf{D}$, based on the previously introduced comparison corpus $\mathcal{W}$. Like in the step of LL key term extraction, the corpus consisting of 100,000 randomly chosen sentences from the German Wikipedia provided by the "Leipzig Corpora Collection" is suitable for this purpose. $\mathbf{D}$ as a second $N \times N$ matrix of co-occurrences is computed from counts in $\mathcal{W}$ and calculation of corresponding Dice statistics. Subtracting of $\mathbf{D}$ from $\mathbf{C}$ delivers a matrix $\mathbf{C}'$ reflecting the divergence of co-

occurrence patterns in the reference collection compared to topic-unspecific language:

$$\mathbf{C}' = \max(\mathbf{C} - \mathbf{D}, 0) \tag{3.11}$$

Values for common combinations of terms (e.g. *Frankfurt Main*) are significantly lowered in $\mathbf{C}'$, while combinations specific to the reference collection remain largely constant. The effect of filtering co-occurrence data in the reference collection is displayed in Table 3.3. Most co-occurrence pairs found in the reference collection $\mathcal{V}$ which also exist in the filter collection $\mathcal{W}$ do not represent the desired context of the research question exclusively. Thus, leaving them out or lowering their contextual statistic measure helps to increase the precision of the retrieval process. Applying the *max* function asserts that all negative values in $\mathbf{C} - \mathbf{D}$ (representing terms co-occurring less significantly together in sentences of the reference collection than in sentences of the filter collection) are set to zero. The remaining co-occurrence pairs sharply represent contexts of interest for the retrieval process (see Table 3.4)

3.1.5. Scoring Co-Occurrences

To exploit co-occurrence statistics for IR, the scoring function in equation 3.9 has to be reformulated to incorporate a similarity measure between a co-occurrence vector profile of each term t in the dictionary and each sentence s in the to-be-scored-document d. In addition to term frequency, we extend scoring by information on contextual similarity of term usage in sentences $s \in d$:

$$\text{tfsim}(t, \mathbf{C}', d) = \sum_{s \in d} \begin{cases} \text{tf}(t,s) + \alpha \times \text{sim}(\vec{s}, C'_{t,\cdot}), & \text{tf}(t,s) > 0 \\ 0, & \text{tf}(t,s) = 0 \end{cases} \tag{3.12}$$

$$\text{sim}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \tag{3.13}$$

The frequency of t within a sentence (which usually equals 1) is incremented by the cosine similarity (see Eq. 3.13) between sentence

Table 3.3.: Examples of Dice statistics for term pairs of which values get drastically lowered in $\mathbf{C}'$ (see eq. 3.11), hence, contributing less to the contextualized relevancy scoring. Co-occurrence statistics from the reference collection ($\mathbf{C}$) which are also observable in the filter collection ($\mathbf{D}$) were ordered by significance ratio between the two collections ($\mathbf{D}/\mathbf{C}$).

a	**b**	**C**	**D**	**D/C**
verlag	auflage	0.046	0.131	2.85
million	insgesamt	0.029	0.048	1.63
demokratisch	partei	0.067	0.104	1.55
weit	verbreiten	0.101	0.155	1.53
verletzen	person	0.025	0.037	1.50
raum	deutschsprachig	0.088	0.129	1.47
jugendliche	kind	0.061	0.090	1.45
geschichte	deutsch	0.022	0.031	1.38
französisch	deutsch	0.022	0.030	1.32
scheitern	versuch	0.109	0.144	1.31
sozial	politisch	0.023	0.030	1.29
vorsitzende	mitglied	0.038	0.049	1.28
iranisch	iran	0.091	0.115	1.26
staatlich	einrichtung	0.054	0.065	1.21
sozialistisch	sozialismus	0.054	0.065	1.20
maßgeblich	beteiligen	0.076	0.088	1.15
september	oktober	0.023	0.026	1.13
million	jährlich	0.048	0.054	1.12
zeitschrift	deutsche	0.026	0.029	1.12
politisch	partei	0.056	0.061	1.08
stellen	fest	0.062	0.067	1.07
staaten	sozialistisch	0.033	0.035	1.07
ziel	erreichen	0.040	0.042	1.04
politisch	mitglied	0.027	0.027	.99
august	juli	0.045	0.045	.99
frage	stellen	0.063	0.062	.98
republik	sozialistisch	0.035	0.034	.95
verschieden	unterschiedlich	0.050	0.047	.93
november	oktober	0.027	0.025	.93

Table 3.4.: Examples of Dice statistics for co-occurrences in $\mathbf{C}'$ after filtering the co-occurrence patterns of the reference collection $\mathcal{V}$ by those from the filter collection $\mathcal{W}$. These term pairs strongly contribute to the contextualized relevancy score.

a	**b**	$\mathbf{C}'$
innern	bundesminister	0.851
grundordnung	freiheitlich	0.707
sicherheit	innere	0.680
hizb	allah	0.666
subkulturell	geprägt	0.577
nationaldemokrat	junge	0.526
motivieren	kriminalität	0.470
inhaftierten	hungerstreik	0.451
unbekannt	nacht	0.441
verfassungsschutz	bundesamt	0.434
sicherheitsgefährdende	ausländer	0.417
wohnung	konspirativ	0.405
nationalist	autonom	0.394
verurteilen	freiheitsstrafe	0.389
sachschaden	entstehen	0.388
unbekannt	täter	0.382
bestrebung	ausländer	0.378
extremistisch	ausländer	0.371
kurdistan	arbeiterpartei	0.365
sicherheitsgefährdende	bestrebung	0.356
orthodox	kommunist	0.324
extremistisch	bestrebung	0.319
fraktion	armee	0.317
sicherheitsgefährdend	extremistisch	0.305
rote	hilfe	0.297

vector $\vec{s}$ (sparse vector of length N indicating occurrence of dictionary terms in s) and the dictionary context vector for t out of $\mathbf{C}'$. Cosine similarity has been proven a useful measure for comparing query vectors and document vectors in the VSM model of IR (Baeza-Yates and Ribeiro-Neto, 2011, p. 76f). Here it is applied to compare usage contexts of terms in sentences from the reference collection $\mathcal{V}$ and sentences of target documents $d \in \mathcal{D}$. Adding contextual similarity to the tf measure rewards the relevancy score, if the target sentence and the reference term t share common contexts. In case dictionary terms occurring in sentences of d share no common contexts, the cosine similarity equals 0 and $tfsim$ remains equal to tf.

Because term frequency and cosine similarity differ widely in their range the influence of the similarity on the scoring needs to be controlled by a parameter α. If $\alpha = 0$, $tfsim$ replicates simple term frequency counts. Values $\alpha > 0$ yield a mixing of unigram matching and context matching for the relevancy score. Optimal values for α can be retrieved by the evaluation method (see Section 3.1.6). Finally, the context-sensitive score is computed as follows:

$$\text{score}_{\text{context}}(q, \mathbf{C}', d) = \text{norm}(d) \times \sum_{t \in q} \frac{1 + \log(\text{tfsim}(t, \mathbf{C}', d))}{1 + \log(\text{avgtf}(d))} \times \text{boost}(t) \tag{3.14}$$

3.1.6. Evaluation

Determining a large set of key terms from a reference collection and extracting its co-occurrence profiles to compose a "query" is an essential step in the proposed retrieval mechanism to meet requirements of content analysts. Due to this, standard approaches of IR evaluation (Clough and Sanderson, 2013) which focus primarily on precision in top ranks and utilization of small keyword sets as queries are hardly applicable. Test collections and procedures such as provided by the TREC data sets (Voorhees, 2005) would need serious adaptions regarding such type of retrieval task (e.g. compiling a reference collection

from the relevant document set). As I also need an evaluation specific to the proposed research question on democratic demarcation, I decided to follow two approaches:

1. Generating a quasi-gold standard of *pseudo-relevant documents* to show performance improvements through the use of co-occurrence data as well as certain methods of key term extraction,

2. Judging on the overall validity manually with *precision at k* evaluation on the retrieved document set for this example study.

Average Precision on Pseudorelevant Documents

To evaluate on *precision* (share of correctly retrieved relevant documents among the top n ranks of a retrieval result) and *recall* (share of correctly retrieved relevant documents among all relevant documents of the collection) of the retrieval process a set of relevant documents has to be defined (Baeza-Yates and Ribeiro-Neto, 2011, p. 135). It is obvious that this set cannot be derived from the collection of newspaper documents investigated in this study, as it is the objective of this retrieval task to identify the relevant documents. Instead, we define a set of 'pseudo-relevant' documents as a gold standard, originating from the reference collection of the "Verfassungsschutzberichte". These annual reports were initially split into 519 pseudo-documents each containing a sequence of 30 sentences ordered by appearance within the reports (see Section 3.1.2). For the evaluation process, I split the reference collection set in two halves:

- 260 pseudo-documents with odd numbering are used for dictionary and co-occurrence extraction,

- 259 pseudo-documents with even numbering are used as gold standard of relevant documents for evaluation.

The retrieval process then is performed on a document set of 20,000 newspaper articles randomly chosen from the FAZ collection and merged with the 259 'gold standard' documents into the evaluation

collection $\mathcal{E}$. To this collection $\mathcal{E}$ the process of relevancy scoring (eq. 3.14) is applied which yields a ranked order of documents. The quality of the retrieval process is better the higher the density of 'gold'-documents in the upper ranks is. This relation can be expressed in precision recall curves. Incorporating results from lower ranks of the scoring into the result set includes more relevant documents which increases recall. At the same time precision of the result set decreases, because more documents not considered as relevant are included as well. Plotted as diagram, different IR systems can be easily compared. The larger the area under the curve, the better the retrieval performance.

Figure 3.1 displays such precision recall curves for different values of α. It shows that using contextualized information ($\alpha > 0$) positively influences the retrieval result compared to simple unigram matching of dictionary terms in documents ($\alpha = 0$). Nonetheless, difference between larger influence of context information ($\alpha = 15$ vs. $\alpha = 30$) seems to be neglectable.

Retrieval quality also can be expressed in a single measure like "average precision" which computes the precision at various levels of recall (Clough and Sanderson, 2013). Figure 3.2 plots average precision for retrieval runs with the three dictionary lists and different alpha parameters. Again, it becomes evident that utilizing contextual information increases retrieval performance, as the precision increases with increasing α values. For $\alpha > 15$ precision does not increase much further. This hints to select $\alpha = 15$ as a reasonable parameter value for the retrieval task to equivalently mix information from unigram matching and context scoring of query terms in target documents.

Furthermore, average precision evaluation allows for comparison of the different term extraction methods used to create the retrieval dictionary from the reference collection. While term extraction via TF-IDF and Topic Models perform almost equal, the dictionary based on Log-likelihood outperforms the other approaches. Obviously this method is more successful in extracting meaningful key terms. This is also confirmed by Figure 3.3 which plots precision-recall curves for the three approaches.

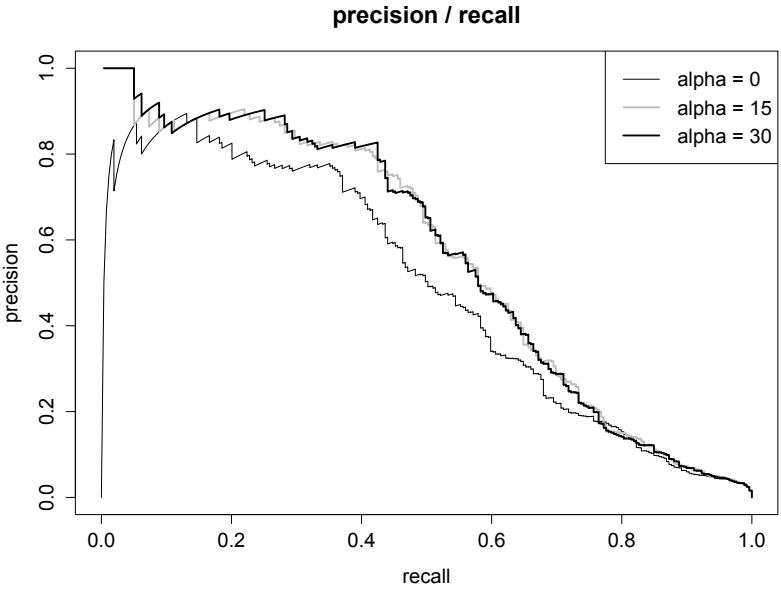


Figure 3.1.: Precision recall curves for contextualized retrieval with Log-Likelihood extracted dictionary and three different α values.

Precision at k

A second evaluation targets directly to the content relevant for the example study. Using the best performing LL dictionary for retrieval in the global newspaper collection $\mathcal{D}$ produces a ranked list of around 600,000 documents. To compare results of context-insensitive matching of dictionary terms with contextualized dictionaries, I ran retrieval twice for $\alpha = 15$ and $\alpha = 0$. For each of the top 15,000 documents per list, I evaluate how dense the relevant documents on different ranges of ranks are. The *precision at k* measure can be utilized to determine the quality of the process by manually assessing the first 10 documents downwards from the ranks 1, 101, 501, 1001, 2501, 5001, 7501, 10001, 12501, 14991 (Baeza-Yates and Ribeiro-Neto, 2011, p. 140). Documents from each rank range were read closely and marked as

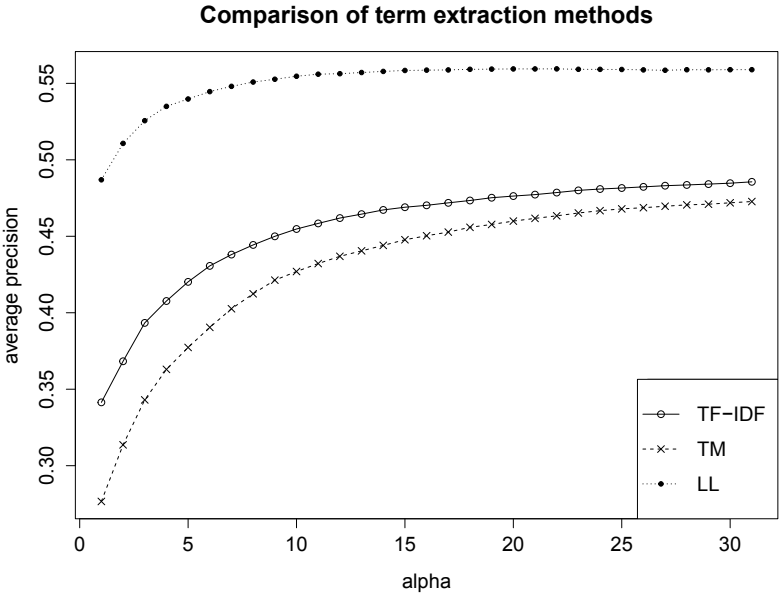


Figure 3.2.: Average retrieval precision for dictionaries based on different term extraction methods and α values.

relevant in case any part of the text expressed a statement towards democratic demarcation. This includes speech acts of exclusion or rebuttal of exclusion of (allegedly) illegitimate positions, actors or activities within the political spectrum.

The results in Table 3.5 confirm the usefulness of the contextualization approach. Density of positively evaluated results in the upper ranks is very high and decreases towards the bottom of the list. Precision in the system utilizing co-occurrence data ($\alpha = 15$) retrieves more relevant documents and remains high also in lower ranks, while it drops off in the system which solely exploits unigram matching between query and document ($\alpha = 0$). Further, since the study on democratic demarcation is targeted to domestic political contexts, I evaluated if retrieved documents were related primarily to foreign

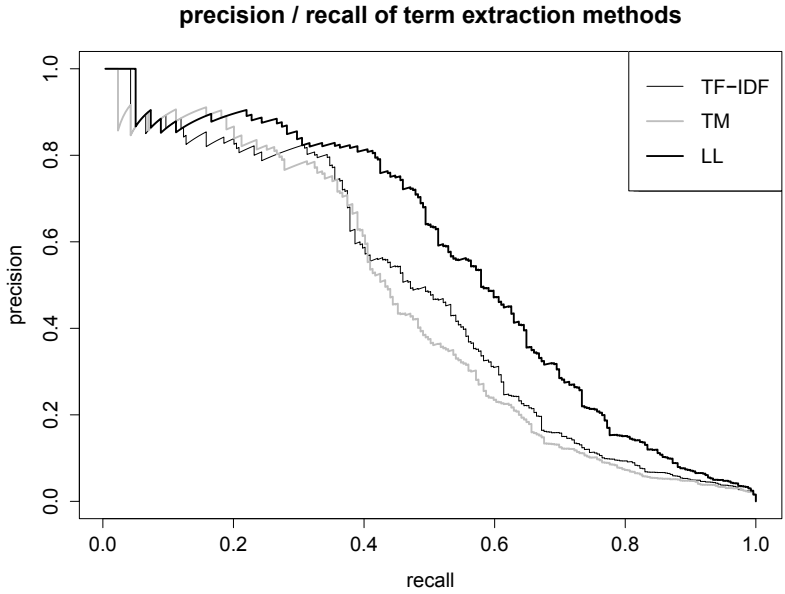


Figure 3.3.: Precision recall curve for dictionaries based on different term extraction methods ($\alpha = 15$).

or domestic affairs. Retrieval with contextualized dictionaries better captured the domestic context from the reference collection of the BfV reports, resulting in a lower share of foreign related documents.

Size of Relevant Document Set

The final retrieval to select (potentially) relevant documents for this example study from the collections of *FAZ* and *Die Zeit* is performed by extraction of a dictionary from the complete reference collection $\mathcal{V}$ of five BfV reports split into 519 pseudo-documents. Corresponding to the evaluation result, the LL approach for term extraction and a retrieval parameter $\alpha = 15$ have been used. Due to the size of the dictionary ($N = 750$ terms), almost every document of both collections gets a relevancy score greater than zero. As a matter of

Table 3.5.: Manually evaluated precision of retrieval results at increasing rank intervals. Both IR systems compared utilize the LL based dictionary.

Precision at k	$\alpha = 0$	$\alpha = 15$
1–10	10	10
101–110	9	8
501–510	7	9
1001–1010	7	9
2501–2510	8	9
5001–5010	5	5
7501–7510	4	5
10001–10010	4	3
12501–12510	2	5
14991–15000	1	4
Total	57	**67**
Foreign related	45%	**34%**

fact, documents in the lower ranks cannot be considered as relevant. The main objective now is to determine a threshold for the relevancy score. Documents below this score would be considered as not relevant. Documents above this score will be the base for the upcoming analysis.

Again, the evaluation approach of the pseudo-gold document set can help to solve this problem. It allows to compute at which rank in our relevancy ordered evaluation collection $\mathcal{E}$, consisting of *FAZ* and BfV documents, a specific recall level of gold documents from BfV reports is achieved. As desired recall level, I strive for around 80 % of all relevant documents. The corresponding rank $r_{0.8}$ can be utilized to estimate a relative proportion of relevant documents in the ranked list of the overall collection. We assume that we look for certain similarities in language use between the reference and the target collections, and that the proposed IR mechanism favors these similarities in its ranking. If this holds true, many documents from

the target collection (which are per definition not part of the 'gold' set in this evaluation) should contain language characteristics similar to the reference collection, and thus, may be considered as relevant for the research question. When applying the retrieval mechanism to comparable collections (e.g. another complete archive of second newspaper), it appears to be a reasonable heuristic to consider the same proportion of the collection as (potentially) relevant as in the (randomly selected) evaluation collection $\mathcal{E}$.

For the evaluation collection $\mathcal{E}$, consisting of 20,259 documents, $r_{0.8} = 1375$ which means that 80 % of the 'gold' documents are located in roughly 7 % of the top ranked documents in the collection ($r_{0.8}/|\mathcal{E}|$). Selecting the top ranked 7 % from the entire FAZ collection ($\approx 200,000$ documents) yields a collection of 14,000 (potentially) relevant documents.[6] Selecting the top ranked 7 % from the entire *Die Zeit* retrieval yields a collection of 28,000 (potentially) relevant documents.

Filtering out Foreign Affairs

Democratic demarcation is not only expressed in the news with regard to domestic affairs. The manually conducted evaluation also showed that lots of documents were retrieved related to foreign affairs (see Table 3.5). Although the proposed IR mechanism decreases the share of documents related to foreign affairs compared to a context-insensitive retrieval, roughly one third of all manually evaluated documents fit in this category. Since this example study is concerned with democratic demarcation in the FRG, I want to filter the retrieval result for documents primarily related to domestic affairs.

This could be formulated as a complex machine learning classification task (see Section 3.3). But for the moment, there is a straightforward base line approach which yields sufficient results, too. For this, I employ two different dictionaries of location entities, either domestic or foreign related:

[6] I decided to use rounded values, because this procedure of determining a traceable threshold is an approximate heuristic.

- *Domestic affairs* (DA): a list of all German federal states and their capitals, and a list of abbreviations of the major parties in the German Bundestag;
- *Foreign affairs* (FA): a list of all United Nations (UN) registered nations and their capitals (except Germany), and a list of all cities over one million inhabitants (except German cities).

Dictionaries of these location entities can be easily compiled from Wikipedia lists which represent a valuable controlled resource for this purpose.[7] These dictionaries are employed to count occurrences of terms they consist of in the retrieved documents. Documents then are categorized by the following rules: documents

- containing at least one FA-term, *and*
- containing less than two DA-terms

are considered to be foreign-related. Evaluation of this 'naive' rule set[8] on the manually evaluated examples shows high values for precision ($P = 0.98$) and recall ($R = 0.83, F_1 = 0.91$).

Documents identified as foreign-related were removed from the retrieval set resulting in the final retrieved collection $\mathcal{D}'$.

3.1.7. Summary of Lessons Learned

As a result of the first analysis task on IR, the retrieved FAZ collection consists of 9,256 documents, the *Die Zeit* collection consists of 19,301

[7] Using a list of states registered at the UN has the advantage that it also includes states that seized to exist (e.g. Czechoslovakia or Yugoslavia).
UN nations:
http://de.wikipedia.org/wiki/Mitgliedstaaten_der_Vereinten_Nationen
Capitals:
http://de.wikipedia.org/wiki/Liste_der_Staaten_der_Erde
Large cities:
http://de.wikipedia.org/wiki/Liste_der_Millionenst%C3%A4dte
FRG states/capitals:
http://de.wikipedia.org/wiki/Land_%28Deutschland%29

[8] For more information on classification evaluation see Section 3.3.5

documents – both mainly containing articles related to domestic affairs and information relevant to the question of democratic demarcation.

Purpose of this task within the overall TM workflow was to identify relevant documents within huge topic-unspecific collections. Furthermore, it should respond to the requirements of 1) identification of relevant documents for abstract research questions, 2) focus on recall to select large sets of relevant documents for further investigation and 3) provide a heuristic solution to decide how many documents to select for further investigation. Lessons learned from approaches to this task can be summarized as follows:

- Compiling a collection of paradigmatic reference documents can be a preferable approach to describe an abstract research interest compared to standard ad-hoc retrieval (Voorhees and Harman, 2000) by a small set of keywords.
- Dictionary extraction for IR query compilation can be realized by key term extraction from the reference collection. The method of LL for key term extraction is preferred.
- Extraction of term co-occurrence statistics from the reference collection contributes to improve retrieval performance over just looking for dictionary terms neglecting any context.
- Average precision based on a pseudo-gold document set compiled from half of the reference collection can be used to automatically evaluate on retrieval performance with respect to an optimal retrieval algorithm.
- Precision at k on final retrieval results can be used to judge manually on quality of the retrieval result with respect to the research question.
- Rankings of documents from the pseudo-gold set can be employed to estimate on proportions of relevant documents in a final retrieval list, providing a heuristic for the number of documents to select.

Table 3.6.: Final data sets retrieved for the study on democratic demarcation.

Corpus	Publication	#Doc	minFrq	#Token	#Type
$\mathcal{D}'_{ZEIT}$	Die Zeit	19,301	10	11,595,578	63,720
$\mathcal{D}'_{FAZ}$	FAZ	9,256	10	2,269,493	20,990
$\mathcal{D}'$	FAZ + Zeit	28,557	15	13,857,289	53,471

- Dependent on the research question, subsequent filter processes on the retrieval results may be useful to get rid of undesired contexts for the QDA purpose, such as domestic versus foreign affairs relatedness of retrieved contents.

Future work on this task could elaborate more closely on the influence of different parameters within the workflow (e.g. altering the dictionary weighting function Eq. 3.5). Moreover, it would be interesting to integrate other sophisticated methods of term weighting and normalization strategies from elaborated ad-hoc approaches of IR to see, if they improve the retrieval quality with respect to the requirements specified.

3.2. Corpus Exploration

The process of document retrieval conducted in Section 3.1 yielded a final collection of (potentially) relevant documents for the further analysis $\mathcal{D}'$ (see Table 3.6). All methods introduced in the following subsections were conducted to explore the combined corpus $\mathcal{D}'$ containing of both publications, *Die Zeit* and *FAZ*, together. The separated inspection of corpora of the single publications is subject of analysis again for classification in Section 3.3. The corpora are preprocessed by the following procedures (see Section 2.2.2 for details on preprocessing):

- tokenization of sentences and terms,

- removal of stop words,
- merging of terms within MWUs to single tokens,[9]
- transformation of named entities to their canonical form,[10]
- lemmatization of tokens,[11]
- lowercase transformation of tokens,
- pruning of all terms below *minFrq* (see Table 3.6) from the corpus.

The pruning of terms below a minimum frequency is a useful step to keep data sizes manageable and data noise due to misspellings and rare words low. For the corpus $\mathcal{D}'$, three different DTMs were computed containing counts of types for each document, for each paragraph per document and each sentence per document separately. Identifiers for sentences, paragraphs and documents allow for selection of corresponding sub-matrices, e.g. all sentence vectors belonging to a document vector. These DTMs are the basis for the second step of unsupervised exploration of the retrieved document collection. Results are shown in this section only for the purpose of exemplary description. A comprehensive description of the interpreted findings during corpus exploration with respect to the research question on democratic demarcation is given in Chapter 4.

3.2.1. Requirements

When investigating large corpora which contain documents of several topics and from different time periods, analysts need methods to

[9] For this, a dictionary of German MWUs was applied which was compiled for utilization in the aforementioned *ePol*-project (Niekler et al., 2014).

[10] For this, a dictionary of variants of named entities assigned to their canonical form was applied. This dictionary is based on the JRC-Names resource provided by the European Commission (https://ec.europa.eu/jrc/en/language-technologies/jrc-names).

[11] For this, a lemma dictionary compiled by the project *Deutscher Wortschatz* was applied (http://wortschatz.uni-leipzig.de).

become familiar with its temporal and thematic contents without reading through all of the material. Sampling a small random subset could help to enlighten certain events and aspects in the newspaper texts. But reading selected sample documents does not give hints on distributions and shares of topics over time—they also do not contribute much to get the "big picture". Instead, one can apply a controlled process of (semi-)automatic, data-driven methods to split the entire collection into manageable segments. These segments should be defined with respect to two dimensions: time and topical structure. Each of the segments then can be described by text statistical measurements, extracted structures of meaning, corresponding graphical visualizations and representative example snippets of text. Knowledge about the overall subject can be derived by investigating and interpreting the segments, each by itself or in contrast with each other.

The procedure proposed in this section can be seen as an implementation of what Franco Moretti has labeled "distant reading" (Moretti, 2007). By combining different methods of NLP, information extraction and visualization, it strives to reveal patterns of meaning in the data. With this, not only fixed manifest contents may be identified for quantification, but also meaningful latent concepts or topics can be identified and their change over time may be tracked. This equips content analysts with an invaluable heuristic tool to grasp knowledge structures and their quantitative distribution within large data sets.

Technically, the proposed procedure may be related to the task of *ontology learning*, which is a sub-field of *ontology engineering* in information management. Ontologies formally describe types of knowledge entities of a certain domain together with their properties and relations. For example, they define a hierarchical set of key terms related to each other by hyponymy, hypernymy, synonymy or antonymy. Such structures can be learned and populated automatically from text collections to a certain extent (Cimiano, 2006). But the conceptualization of ontologies in the field of information management differs from the application requirements in QDA with respect to several aspects. Definitions of ontologies in information systems are built in a

very formal way to capture characteristics of a single domain with the objective to unify knowledge application in an intersubjective manner. Knowledge representations might even be machine readable in a way that allows for logical reasoning. This is a key requirement to support business processes or knowledge bases for applications of artificial intelligence. In contrast to this, content analysts (especially in discourse analysis) are interested in knowledge patterns spread around multiple domains with rather "soft" characterizations of concepts. If longer time periods come into play they even have to deal with changes of meaning instead of fixated definitions. Moreover, especially those concepts are interesting for investigation which appear as intractable to intersubjective formal definition—so called "empty signifiers" (Nonhoff, 2007, p. 13) such as democracy, extremism, freedom or social justice which may be understood in countless manifold ways, but hardly fit into a formal ontology. We might even say that the field of information systems and the field of social science seem to represent opposing ends concerning their epistemological fundamentals—the former strives for fixation of structures to model representations of essential beings while the latter strives for *re*-construction and understanding of elusive knowledge structures shaped by discursive language evolvement over time. Luckily, this difference only is important for the application of extracted knowledge. For identification of structures and patterns both profit from a variety of data-driven NLP technologies.

For knowledge extraction, I can rely on approaches proven as useful for ontology learning, especially the idea to combine several NLP techniques to create data-driven excerpts of knowledge patterns from document collections. But instead of putting extracted information into formal conceptual containers with strict relations for my intended purpose, it is advised to provide an intuitive access to the data which allows researchers to inductively explore meaningful knowledge patterns. This intuitive access is provided by graphical display of co-occurrence networks of semantically coherent key terms in temporal and thematic sub-parts of the corpus. For generating such network graphs, I combine several technologies:

- a topic model based on the entire corpus,
- a clustering in time periods based on topic distributions,
- a co-occurrence analysis of terms per topic and time frame,
- a 'keyness' measure of significantly co-occurring terms,
- a dictionary-based sentiment measure of the key terms,
- a heuristic to identify semantic propositions based on maximal cliques in co-occurrence networks, and
- a method to extract representative example text snippets based on propositions.

The following subsections describe these methods of information extraction, pattern recognition and computation of textual statistics. Finally, information generated by the largely automatic processes is utilized as an input to render informative graphs which I call Semantically Enriched Co-occurrence Graphs (SECGs)—one for each topical cluster within a distinctive time frame. SECGs allow for visual intuitive investigation of interesting content parts within large document collections. While processes are largely unsupervised and data-driven, analysts still have to make conscious decisions in some steps in order to control parameters of processes or to make manual selections of intermediate results. This should not be seen as a flaw of the entire process chain, but as an opportunity to keep control over the analysis and to get a deeper understanding of the data and methods used.

3.2.2. Identification and Evaluation of Topics

The to-be-explored corpus $\mathcal{D}'$ contains 28,557 retrieved newspaper articles published over six decades. Figure 3.4 shows that documents are unequally distributed over time. While there are less than 200 documents per year in the beginning of the time period investigated, their number increases in the early 1960s. The smoothed long-term development shows three peaks around 1968, 1990 and 2001. We can

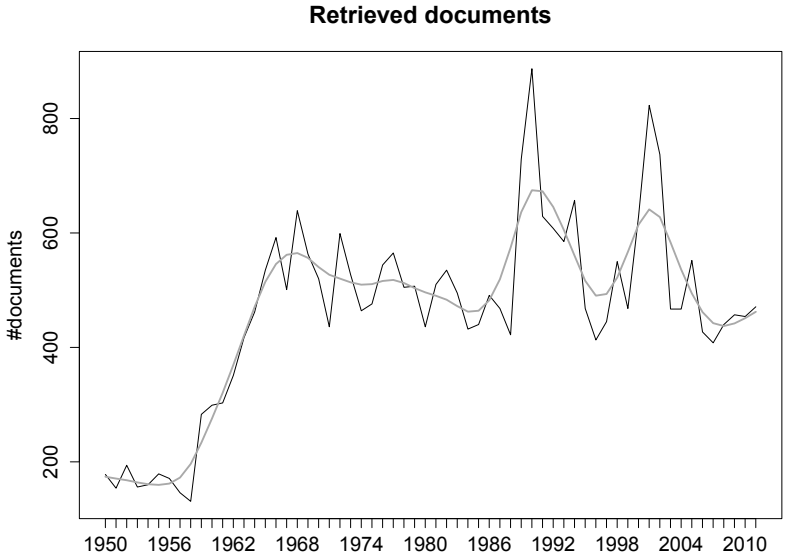


Figure 3.4.: Retrieved documents from *Die Zeit* and *FAZ* over time. The grey line represents a smoothed spline showing three peaks in long term development of the document distribution. This collection will be explored by data-driven methods in Section 3.2.

assume that certain discursive events related to democratic demarcation are responsible for these trends which can be traced by topic changes.

To support exploratory investigation of the collection $\mathcal{D}'$, topic models provide a valuable approach to cluster thematic contents. Topic models allow for investigating contents through a 'distant' perspective by inference of term distributions $\beta_{1:K}$ in K topics representing semantically coherent clusters of term usage, and inference on topic distributions $\boldsymbol{\theta}$ in documents. Topic-document distributions can be observed in single documents or in aggregated sub-collections, e.g.

documents from selected time frames. Furthermore, as documents are modeled as a mixture of topics, computed model instances allow for collection filtering on the basis of presence of inferred latent topic structures above a certain threshold. This contextual filtering is an essential step to generate thematic sub-collections on which further text statistical measures can be applied. Expressiveness of such measurements dramatically increases if a largely coherent context of the underlying collection through topic filtering can be guaranteed.

Model and Parameter Selection

For the purpose of topic identification, I rely on the standard parametric LDA model (Blei et al., 2003)—for reasons of simplicity[12] and because I prefer to keep control over the number of topics K to be inferred by the model. Besides K, parametric[13] topic models are governed by hyperparameters influencing the quality of its outcome. Hyperparameters are settings of the prior distributions in topic models. For LDA, the topic distribution per document $\boldsymbol{\theta}$ is determined by a prior α and the term distribution per topic $\boldsymbol{\beta}$ is determined by a prior η. Although it is possible to optimize these parameters automatically for model selection, selecting 'good' settings in QDA scenarios is a rather intuitive process which should be taken out carefully by analysts. Usually, in NLP developers of topic models evaluate their models in automated processes while in QDA scenarios analysts compute models with different parameter settings and judge on outcomes by manual investigation (Evans, 2014). For automatic evaluation, data sets can be divided into one part for model computation and another part of

[12] I used the performant implementations of LDA and CTM provided as packages for R by Grün and Hornik (2011).

[13] Numerically optimal choices for K can be retrieved automatically by non-parametric topic models such as HDP-LDA (Teh et al., 2006) which are reported to deliver slightly better topic qualities. But, loosing control over deliberate selection of K means giving up control over topic granularity which is in my view a relevant parameter in hands of the QDA analyst. Nonetheless, experiments with non-parametric or even time-dynamic topic models for QDA might be an interesting extension to the base line I present here.

model evaluation. The quality of the model is assessed by computing its *perplexity*, i.e. a metric based on the probability of the documents held out for evaluation. Hyperparameter settings then can also be optimized according to highest held out likelihood (Wallach et al., 2009). Although likelihood evaluation is widely used due to its pure automatic nature, Chang et al. (2009) have proven with large user studies that optimal held out likelihood does not correspond to human perception of semantic coherence of topics. My experiments with LDA and computationally optimized hyperparameters as well as with the Correlated Topic Model (CTM) (Blei and Lafferty, 2006) to generate SECG confirmed this finding. Topics of likelihood optimized models on the *FAZ* and *Die Zeit* data were less expressive and less targeted to specific semantic units than topics computed with models of some of my manual parameter selections. Numerical optimization of α values estimated higher optimal values leading to topic distributions in documents where many topics contribute a little probability mass—in other words, topics were less distinct. This diminishes the usefulness of the model for document selection as well as for identification of specific discursive event patterns over different time frames. The CTM model, although in model evaluations yielding better results in terms of likelihood of the data, inferred term distributions of which most topics were dominated by high frequent terms reducing the perceived specificity of these topics to describe a semantic coherence.

Mimno et al. (2011) responded to this circumstance by suggesting a new evaluation metric for topic models. They measure *coherence* C of a topic k by observing co-occurrences of the top N terms of each topic on a document level (ibid., p. 265):

$$C(k, V^k) = \sum_{n=2}^{N} \sum_{l=1}^{n-1} log \left(\frac{D(v_n^k, v_l^k) + 1}{D(v_l^k)} \right) \tag{3.15}$$

Hereby, $V^k = (v_1^k, \cdots, v_N^k)$ represents a list of the N terms of topic k with highest probability. $D(v, v')$ is the frequency of co-occurrence of the types v and v'. Basically, it favors models putting more probability weight on terms in one topic which actually are co-occurring

in documents. Mimno et al. show that their metric is superior to log likelihood in terms of correspondence to user evaluation on the quality of topic models. Furthermore, as the purpose of this subtask is to generate co-occurrence graphs for corpus exploration, topic coherence appears to be the measure of choice for optimization of hyperparameters.

Nevertheless, numerical evaluation measures should not be the single criterion for model selection. First published empirical studies using topic models for QDA also strongly rely on judgments by the human researcher. According to Evans (2014), a conventional procedure is to compute a variety of different models with different parameters. These models then are compared by the analyst with respect to the question which one fits best to the research objective. This validation can be done in three steps:

1. investigating the top N most probable terms of each topic and check if it is possible to assign a descriptive label to them,

2. comparing measurements of semantic coherence of topics (see eq. 3.15) as an additional hint to identify overly broad or incoherent topics, and

3. evaluating whether topic distribution over time follows assumptions based on previous knowledge of the researcher (e.g. if topics on Islam and terrorist activities in the news co-occur in the 2000s, but not before 2001).

I have performed model selection by combining numeric optimization based on the topic coherence measure with steps of the manual evaluation procedure. Firstly, I decided for using $K = 100$ as a satisfying topical resolution for the collection. It is possible to judge on 100 topics manually, but the number is still high enough to capture also smaller thematic patterns which might play a role only in shorter periods of time of the overall discourse on democratic demarcation. Secondly, I computed six LDA models with different settings of α.

Each model computation was carried out with 1,000 iterations of Gibbs Sampling using the following parameters:[14]

- pruning of all types that appear less than 35 times in the corpus to reduce data size for model computation which left 30,512 types,
- $K = 100$ topics, $\eta = 10/K = 0.1$[15]
- $\alpha \in \{0.05, 0.1, 0.2, 0.3, 0.5, 1.0\}$[16]

The objective is to identify one model which yields a mixture of rather few topics with high probability allowing for more specific contextual attribution than a mixture of many broader topics. Thus, for each of the six models its mean topic coherence $(\sum_{k=1}^{K} C(k, V^k))/K$ was computed using the $N = 100$ most probable terms per topic. The results in Figure 3.5 indicate that the model with $\alpha = 0.2$ achieves highest coherence. It also shows that likelihood of the models does not correlate to their coherence measure. In the last step, a manual investigation of the topic defining terms suggested that in fact the model computed with $\alpha = 0.2$ provided the most coherent and descriptive topics for the research questions.

Performing all steps for model selection, I chose the model $K = 100, \eta = 0.1, \alpha = 0.2$ as the basis for the upcoming steps. The inferred topics of this model are displayed in Table 3.7 by their most probable terms and their distribution θ_k in the entire corpus $\mathcal{D}$.

[14] Every model took roughly 9 hours computation time using the implementation provided by Grün and Hornik (2011).

[15] For the η prior, I stuck with the default value of the topic model implementation, since I am primarily concerned with topic specificity to describe document contents governed by α.

[16] Lower α priors lead to inference of fewer, more specific topics determining document contents. For higher α priors, documents are modeled as mixtures of more evenly distributed topics.

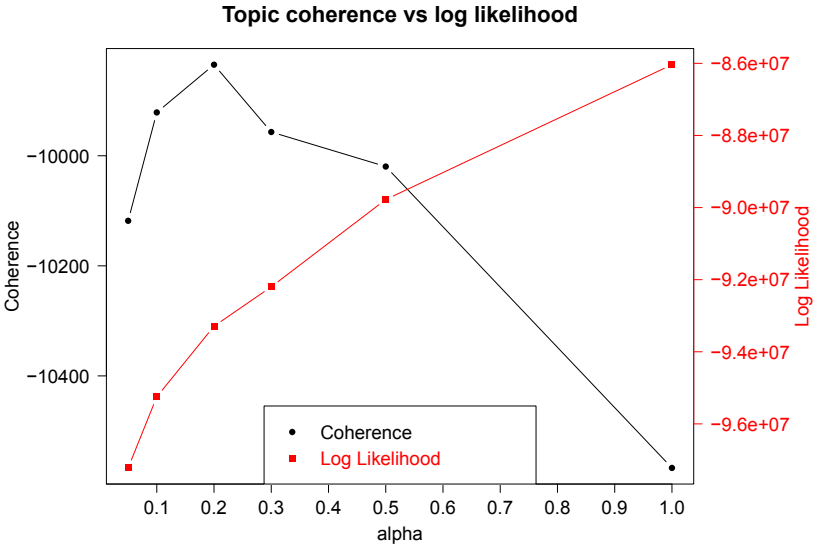


Figure 3.5.: Mean topic coherence and log likelihood of all six topic models computed with different α values. The model with $\alpha = 0.2$ achieved highest coherence.

Table 3.7.: Topics in $\mathcal{D}'$ ordered by rank_1: for each topic, the table displays ID, top eight terms, proportion within the corpus (θ_k), number of documents where it has maximum share of all topics (C_{r_1}), rank according to topic probability (r_P) and rank_1 (r_1).

ID	Top terms	θ_k	C_{r1}	r_P	r_1
71	spd partei wähler cdu wahl prozent wahlkampf fdp	0.0176	730	1	1
3	zeit glauben frage herr mensch leute jahr groß	0.0166	581	4	2
35	jahr leute haus alt tag leben stadt sitzen	0.0163	560	5	3
41	buch politisch geschichte band autor verlag darstellung beitrag	0.0143	538	12	4
68	europa europäisch gemeinsam politisch europäischen gemeinschaft national	0.0121	531	20	5
88	militärisch europa nato sowjetunion staaten bündnis westlich politisch	0.0123	521	19	6
91	fdp liberal partei genscher koalition demokrat freier westerwelle	0.0096	507	50	7
58	gewerkschaft arbeitnehmer arbeiter dgb streik organisation mitglied betrieb	0.0092	501	59	8
83	krieg international militärisch deutschland nation einsatz nato vereint	0.0094	478	55	9
32	spd schröder lafontaine partei kanzler gerd_schröder müntefering rot-grün	0.0102	475	41	10
90	ddr sed ulbricht sozialistisch honecker partei kommunistisch sozialismus	0.0105	471	39	11
51	npd republikaner rechtsradikal rechen gewalt jahr rechtsextrem neonazi	0.0089	465	62	12
59	bundesrepublik deutschen ddr wiedervereinigung deutschland anerkennung	0.0114	449	27	13
76	richter gericht urteil justiz jahr angeklagte verfahre politisch	0.0102	426	42	14
62	partei dkp kommunistisch politisch kommunist verbot öffentlich dienst	0.0095	417	53	15
54	erklären regierung französisch außenminister amerikanischen usa deutschland	0.0096	416	49	16
13	student universität hochschule schule professor schüler lehrer bildung	0.0085	413	68	17
63	polizei demonstration gewalt demonstrant aktion protest polizist student	0.0095	383	52	18
42	terrorist terrorismus anschlag raf mord terror gruppe terroristisch	0.0088	379	64	19
11	merkel angela_merkel union partei cdu kanzlerin koalition koch	0.0096	376	51	20
4	volk mensch welt leben groß kraft freiheit zeit	0.0141	375	13	21
87	spd partei sozialdemokraten brandt sozialdemokratisch wehner vogel willi_brandt	0.0105	374	38	22
50	jahr land sozial prozent milliarde arbeitslosigkeit hoch steuer	0.0111	373	29	23

10	grundgesetz verfassung land gesetz artikel bundestag bundesverfassungsgericht	0.0118	371	21	24
73	pds partei spd gysi osten linkspartei land sachsen	0.0075	371	77	25
85	kanzler regierung opposition schmidt koalition adenauer bundeskanzler groß	0.0112	370	28	26
74	ddr weste osten einheit alt ostdeutsch bundesrepublik vereinigung	0.0091	360	60	27
95	ausländer deutschland flüchtling land bundesrepublik deutschen jahr deutsche	0.0080	357	72	28
65	verfassungsschutz polizei behörde information geheimdienst wissen beamte	0.0093	355	58	29
78	deutschland adenauer deutschen zone westlich deutschlands deutsche weste	0.0098	353	46	30
20	grüne grün fischer partei grünen ökologisch jahr kernenergie	0.0081	349	71	31
29	hitler deutschen reich nationalsozialismus widerstand deutsche krieg	0.0088	346	63	32
79	frankreich französisch paris italien gaulle frankreichs italienischen franzosen	0.0083	343	70	33
14	land jahr wirtschaftlich prozent bundesrepublik wirtschaft groß industrie	0.0109	339	33	34
94	schriftsteller buch literatur jahr roman schreiben literarisch autor	0.0075	338	79	35
44	politik groß frage jahr republik kanzler denken müssen	0.0107	335	34	36
43	partei parteitag vorsitzende delegierte mitglied wahl parteivorsitzende wählen	0.0127	334	17	37
46	politisch politik gegenüber stark gewiß groß werden freilich	0.0176	324	2	38
96	türkei türkisch islam türke muslime islamisch deutschland muslimisch	0.0059	323	93	39
52	sozial gesellschaft mensch politik staat leben freiheit arbeit	0.0105	317	37	40
17	sozialismus sozialistisch revolution kommunistisch kommunist marx kapitalismus	0.0101	313	43	41
33	jahr leben werden tod freund gefängnis verhaften zeit	0.0098	307	45	42
86	iran arabisch afghanistan irak land iranisch islamisch welt	0.0066	300	83	43
12	politisch gesellschaft gesellschaftlich system sozial gruppe entwicklung form	0.0139	299	14	44
77	kirche katholisch christlich evangelisch christ bischof kirchlich gott	0.0066	298	84	45
100	frage aufgabe einzeln groß möglichkeit notwendig gebiet öffentlich	0.0171	297	3	46
48	cdu kohl union partei helmut_kohl schäuble biedenkopf politisch	0.0088	295	65	47
53	mark million geld partei jahr spende stiftung zahlen	0.0080	288	73	48
19	müssen sein können werden politik frankfurter bundesregierung zeitung	0.0143	286	11	49
80	zeitung journalist fernsehen medium rundfunk presse blatt programm	0.0075	286	76	50
6	jahr politisch freund groß halten lassen leben persönlich	0.0145	278	10	51
2	land welt afrika hilfe jahr international entwicklungsland entwicklungshilfe	0.0065	269	86	52

7	film kunst sport künstler ausstellung theater spiel bild	0.0064	269	88	53
99	mensch leben welt wissen geschichte glauben volk wahrheit	0.0134	267	16	54
39	abgeordnete parlament bundestag partei wahl fraktion mehrheit stimme	0.0101	261	44	55
57	jude jüdisch israel antisemitismus israelisch deutschland antisemitisch holocaust	0.0055	259	97	56
66	csu strauß bayer bayerisch stoiber münche franz-josef_strauß münchner	0.0062	255	92	57
16	ungarn land tschechoslowakei prag kommunistisch rumänien jugoslawien	0.0063	253	90	58
40	pole polnisch polen warschau polnischen deutschen jahr grenze	0.0053	250	98	59
47	land cdu ministerpräsident hessen nordrhein-westfalen spd landtag niedersachsen	0.0094	250	54	60
64	kritik meinung werden öffentlich frage vorwurf politisch brief	0.0155	246	8	61
93	hitler deutschen weimarer_republik republik reich reichstag deutsche weimar	0.0065	246	85	62
9	≫die jahr deutschland internet ≫wir ≫ich gut berlin	0.0097	237	48	63
67	deutschen revolution bismarck preußisch preuße deutschland könig groß	0.0068	233	82	64
92	wirtschaft sozial marktwirtschaft staat wirtschaftspolitik unternehmer ordnung	0.0084	232	69	65
84	amerika amerikanischen vereinigten staaten amerikanische usa präsident	0.0075	229	80	66
23	bonn bundesrepublik beziehung deutschen bundesregierung gespräch politisch	0.0110	227	30	67
98	frau kind jahr jugendliche jung familie mann jugend	0.0089	224	61	68
30	nation national geschichte kultur kulturell politisch deutschen europäisch	0.0093	222	57	69
81	mitglied organisation gruppe verband verein gründen jahr arbeit	0.0105	222	36	70
31	regierung land präsident jahr volk wahl million bevölkerung	0.0086	221	67	71
70	staat freiheit recht bürger demokratisch gesetz staatlich ordnung	0.0123	212	18	72
25	politisch gesellschaft öffentlich lassen bild debatte öffentlichkeit gerade	0.0114	209	26	73
89	sowjetunion moskau sowjetisch stalin sowjetischen rußland kommunistisch	0.0076	208	75	74
61	china chinesisch land japan peking jahr chinese welt	0.0053	207	99	75
72	rede wort sprechen tag beifall saal sitzen stehen	0.0115	207	24	76
45	frage lassen gewiß bundesrepublik beispiel grund scheinen gut	0.0161	203	7	77
27	unternehmen firma bank jahr geld wirtschaft markt groß	0.0079	196	74	78
55	prozent jahr zahl bevölkerung bundesrepublik million hoch groß	0.0103	193	40	79
1	vertrag staaten international verhandlung beziehung gemeinsam regierung frage	0.0107	190	35	80
82	intellektuelle denken philosophie welt theorie gesellschaft philosoph mensch	0.0075	188	78	81

36	wissenschaft institut wissenschaftler forschung professor international jahr	0.0064	181	89	82
60	partei politisch politik groß wähler programm volkspartei mitglied	0.0115	178	25	83
49	bundeswehr soldat militärisch armee general offizier truppe krieg	0.0063	177	91	84
75	lassen freilich langen rechen müssen woche bleiben gewiß	0.0162	176	6	85
24	minister amt beamte politisch ministerium staatssekretär dienst öffentlich	0.0087	174	66	86
26	berlin berliner stadt politisch berlins hauptstadt west-berlin bürgermeister	0.0065	173	87	87
8	kris regierung politisch reform land groß lage zeit	0.0155	167	9	88
34	britisch england land großbritannien regierung london britischen brite	0.0057	164	96	89
15	demokratie demokratisch politisch bürger volk system regierung parlament	0.0094	156	56	90
97	politisch ziel wichtig entwicklung aufgabe führen gemeinsam diskussion	0.0138	149	15	91
21	krieg frieden weltkrieg groß rußland deutschland spanien politik	0.0071	148	81	92
37	bundespräsident präsident amt politisch weizsäcker wahl kandidat rau	0.0059	143	95	93
69	hamburg hamburger stadt bürgermeister jahr breme bremer politisch	0.0059	142	94	94
5	politisch politik politiker entscheidung handeln frage moralisch bürger	0.0116	105	23	95
56	linke politisch link konservativ radikal mitte liberal rechen	0.0098	103	47	96
22	österreich schweiz österreichisch wien schweizer land jahr österreicher	0.0039	102	100	97
18	deutschen deutschland deutsche deutscher land deutschlands bundesrepublik	0.0116	92	22	98
38	jahr geschichte zeit groß alt bundesrepublik jahrzehnt siebziger	0.0109	66	32	99
28	tag juni november jahr oktober mai september märz	0.0109	60	31	100

Topic Model Reliability

Since the number of possible topic structures in LDA and other topic models is exponentially large, exact solutions for the models are computationally intractable (Blei, 2012, p. 81). Therefore, topic models rely on sampling-based or variational algorithms to find approximate solutions. The Gibbs sampler I used in this study, employs Markov chains as a random process to find a posterior distribution of model parameters close to the true posterior. Unfortunately, the state space of topic models consists of numerous local optima. As a consequence, the inference mechanism not only infers slightly different parameter values each time the algorithm runs for a sequence of finite sampling iterations. If the data is not separable well by the given number of topics K, solutions also may differ widely in terms of underlying latent semantics captured by the inferred topics. This can lead to low reproducibility of a model between repeated runs of the inference algorithm which may question the usefulness of the model for social science goals (Koltcov et al., 2014). To evaluate on reproducibility, Niekler (2016, p. 137f) introduces a procedure to match most similar pairs of topics from two different model inferences by cosine distance (see Eq. 3.13) between their topic-word distributions above a certain threshold t. Since most of the probability mass of a topic is concentrated at only a fraction of the vocabulary, it is suggested to only incorporate the N most probable words from each topic to calculate distances. Practically, this procedure resembles manual evaluation steps human coders apply to decide on similarity between two topics—they also look at the list of the most probable topic words and compare, how similar they are to each other. A high intersection of shared terms indicates that the same topic label could be applied to them.

For evaluating reproducibility of the previously computed model, I repeated the model inference on $\mathcal{D}'$ with the selected parameters five times. Then, I applied the matching procedure on the $N = 100$ most probable terms per topic and with a maximum distance $t = 0.3$ to find pairs between topics from all possible pairs of models. Since there are $i = 5$ models, we can compare matchings for $\binom{i}{2} = 10$ pairs. The mean

number of topic pairs matched from each of the the 10 model pairs gives a measure for the reliability of the model computation on our target collection. On average, 80.7% of the topics could be matched between several model inferences. This is a quite acceptable measure of reproducibility in the context of content analysis, particularly because the matching is successful for the most prominent topics capturing the largest share of the collection. Even when restricting the distance criterion to a threshold of $t = 0.2$, still 70,0% of the topics can be matched. If reproducibility had been insufficient due to bad separability of the investigated collection, it would have been advisable to change model parameters, at first lowering the number of topics K, or apply further measures to increase the reliability.[17]

3.2.3. Clustering of Time Periods

When exploring large document collections, it is helpful to split these collections not only thematically, but also in their temporal dimension (Dzudzek, 2013; Glasze, 2007). Identification of varying time periods allows for embedding analysis results in different historical contexts (Landwehr, 2008, p. 105). This is important because knowledge structures and semantic patterns change substantially over time by mutual influence on these contexts. Thus, gaining insights in long term developments of discourse considerably profits from observations of such changes by comparing different sub-collections split by time. Two strategies can be applied to achieve a temporal segmentation of a diachronic corpus. Time periods can be segmented manually based on text external theory-driven knowledge, e.g. legislative periods or crucial discursive events. They also can be segmented in a data-driven manner by looking for uniformity and change in language use of the corpus. For this, contents can be aggregated according to time slices to be subject of a cluster analysis.

[17] Lancichinetti et al. (2015) proposed Topic Mapping—a method to increase reproducibility by initializing topic-word assignments deterministically based on co-occurrences of words before sampling.

I apply such a data-driven clustering approach for temporal segmentation of $\mathcal{D}'$ on single years of news coverage. Such a clustering on newspaper data from longer time periods reveals clusters of mostly ongoing year spans. For the upcoming steps of analysis, this procedure helps to segment time spans for contrasting investigations. For the generation of SECGs it is useful to allow for graph representations of single thematic contents within a specific period of time, as well as for comparison of same thematic coherence across time periods.

In the previous section, we split the collection on democratic demarcation in different mixtures of topics. These topic distributions may also be utilized to identify time periods which contain similar mixtures of topics. A multitude of other measurements could also be employed to cluster time periods. For example, aggregating word counts of every document in a single year could serve as a basis for creating a *year-term-matrix* analogue to a DTM which serves well as a basis for clustering. But, using topic model probability distributions has the advantages that they 1) are independent of the number of documents over time, 2) have a fixed values range, 3) represent latent semantics, and 4) we already have computed them.

For clustering of years, we average document topic probabilities $\theta_{d,\cdot}$ from all documents published in year $y \in Y = (1950, 1951, \cdots, 2011)$ in the corpus:

$$\theta_{y,k} = \frac{1}{|\mathcal{D}'_y|} \sum_{d \in \mathcal{D}'_y} \theta_{d,k} \tag{3.16}$$

where $\mathcal{D}'_y$ is a subset of all documents from $\mathcal{D}'$ published in year y. This results in a $|Y| \times K$ matrix of topic probability distributions for each year. This matrix has to be transformed into a $|Y| \times |Y|$ distance matrix, representing dissimilarity of topic distributions between all pairs of years, which serves as the basis for a clustering algorithm. Since we deal with probability distributions, Jensen–Shannon Divergence (JSD) is a reasonable choice as distance metric for this

purpose.[18] According to Endres and Schindelin (2003), the square root of JSD should be applied when using it as distance. Thus, for each pair of years we compute:

$$\text{dist}(y, y') = \sqrt{\text{JSD}(\theta_{y,\cdot}, \theta_{y',\cdot})} \tag{3.17}$$

$$\text{JSD}(x, y) = \frac{1}{2}\text{KL}(x : \frac{1}{2}(x + y)) + \frac{1}{2}\text{KL}(y : \frac{1}{2}(x + y)) \tag{3.18}$$

using the Kullback-Leibler divergence $KL(x : y) = \sum_i x_i \log(\frac{x_i}{y_i})$. The distance matrix computed this way can be used as input for any clustering algorithm.

Model and Parameter Selection

Clustering algorithms can be distinguished into nonparametric and parametric approaches: the former decide on a suitable number of clusters based on the data, the latter fit to a given number of clusters k. Although nonparametric approaches appear to be attractive for identification of time periods in diachronic corpora, not all of such algorithms are suitable for clustering on text data. As language data from ongoing year spans is changing rather gradually, common nonparametric density-based clustering algorithms are hardly suitable for the problem of separating clusters of years. DBSCAN (Ester et al., 1996), for example, produces without proper tweaking one big cluster, due to the fact that vector representations of years usually fulfill the properties of density-reachability and density-connectedness in the vector-space under investigation. More suitable are parametric clustering approaches which assign data points to a previously defined number of k clusters, e.g. k-means, k-medoids or Partitioning Around Medoids (PAM). The proper selection of k then can be heuristically supported by a numerically optimal solution determined by a cluster quality index. Yet, the flexible choice of k also provides an opportunity

[18] JSD is commonly used for comparing topic model posterior probability distributions. For example, in Dinu and Lapata (2010); Niekler and Jähnichen (2012); Hall et al. (2008)

for the content analyst to control the number of clusters independent from numerical optimization. This can be a reasonable requirement because of the number of time periods for further investigation should not exceed a certain threshold—from the analyst's perspective it may seem intuitively better to investigate three or four periods within half a century rather than the numerically optimal solution of 20 or more clusters. For these reasons, I decided to use the PAM algorithm (Reynolds et al., 2006) for clustering of time periods.

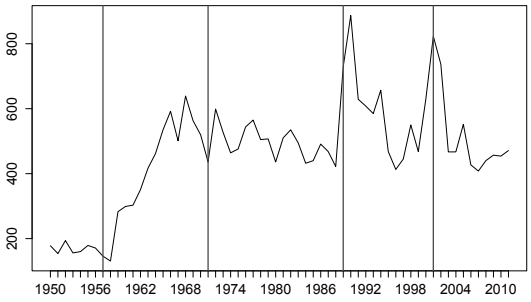
The PAM algorithm clusters observations similar to the famous k-means algorithm by assigning data points to a nearest cluster center. In contrast to k-means, cluster centers are not represented by means of all assigned observations. Instead, PAM uses k medoids, i.e. cluster representative data points from the set of observations. Because no cluster means have to be calculated, PAM can run faster than k-means. But even more important, it runs with an initial 'build phase' to find the optimal medoids for initialization of the algorithm. This does not only lead to faster convergence, but also results in entirely deterministic clustering.[19]

PAM needs a previously defined number of clusters k to run. For content analysis, it is hard to define in advance how many clusters one should expect. We can certainly assume some upper boundary. For manual investigation of the to-be-generated SECGs, it would be hard to split the data into more than 10 time periods. Hence, we may employ a data-driven measurement to determine a value for $k \in \{2, \cdots, 10\}$ yielding optimal separation of the data. I decided for the Calinski-Harabasz index (Caliński and Harabasz, 1974) which is widely used as a heuristic device to determine cluster quality. For each k the clustering is done and for its result the CH-index is computed. As Figure 3.6 shows, the optimal CH-index is achieved when dividing the course of years into five clusters. Running PAM with $k = 5$ yields the time periods presented in Table 3.8.

[19] k-means in contrast may produce different clustering results, if the data is not well separable. Due to random initialization of the cluster means at the beginning, the algorithm it is not fully deterministic.

Figure 3.6.: Calinski-Harabasz index for clusterings with different k. $k = 5$ can be assumed as an optimal solution slicing the news coverage between 1950 and 2011 in five time periods.

Table 3.8.: PAM clustering on topic probabilities per year results in five ongoing year spans over time.

Cluster / years	Docs	Distribution over time
1. 1950–1956	1192	
2. 1957–1970	5742	
3. 1971–1988	8852	
4. 1989–2000	7068	
5. 2001–2011	5703	

3.2.4. Selection of Topics

Not all inferred $K = 100$ topics play an important role in every cluster over time and not all highly probable topics in a single cluster are relevant for the research questions. Thus, for corpus exploration we need a deliberate selection how many and which topics to investigate further. I decide to concentrate on the $K' = 10$ most important topics for each cluster. But how can they be selected? To identify topics relevant for the research question within a certain time frame, we first need a ranking of the topics. From this ranking, we then can manually select K' topics per time frame from the top downwards. Again, manual selection should be seen as an opportunity for the analyst to control the overall process with respect to her/his research objective, rather than a deficiency in a pure data-driven process. Of course, it would also be possible to select automatically just the K' top ranked topics. But chances are high that, on the one hand, we include overly general topics which are not related directly to the research question and, on the other hand, miss important, but rather 'small' topics.

Ranking Topics

For each topic its overall proportion θ_k in the entire corpus $\mathcal{D}'$ can be computed by averaging of all topic shares $\theta_{d,k}$ of each document d:

$$\theta_k = \frac{1}{|\mathcal{D}'|} \sum_{d \in \mathcal{D}'} \theta_{d,k} \tag{3.19}$$

Accordingly, topics can be ordered by their share of the entire collection. It can be found that distribution of topics has similarities to distribution of words in general language: The most probable topics within a corpus are not necessarily the most meaningful topics.[20] The two most probable topics #71 and #46 consist of rather general terms like *partei, wahl, prozent* and *politisch, politik, groß, werden* which

[20] Characteristics of term distributions in natural language can be formally described by Zipf's law (Heyer et al., 2006, p. 87).

do not describe a single coherent theme, but account for relevant vocabulary in many documents concerned with various topics around politics. The same can be diagnosed for the other eight topics #100, #3, #35, #75 #45, #64, #8, #6 of the top ten ranked by probability (see Table 3.7).

To order topics in a more expressive manner, the *rank_1* measure can be applied. For this, we count how often a topic k is the most prominent topic within a document:

$$C_{r1}(k) = \sum_{d \in \mathcal{D}'} \begin{cases} 1, & \forall j \in \{1, \ldots, K\}\ \theta_{d,k} >= \theta_{d,j} \\ 0, & \text{otherwise.} \end{cases} \tag{3.20}$$

The normalized measure $C_{r1}(k)/|\mathcal{D}'|$ expresses the relative share of how often a topic k has been inferred as primary topic. This provides a basis for ranking all topics with respect to their significance on a document level. Topics are ordered by rank_1 in Table 3.7 as well as in Figure 3.7. The figure illustrates the effect of the re-ranking by primary topic counts on the document level. Topics with high share in the entire corpus might be distributed rather evenly over many documents without constituting a proper theme for a document by themselves. These topics are identifiable by the higher bars throughout the lower ranks of the plot.

Manual Selection

The top 25 topics of each temporal cluster ranked by rank_1 are investigated manually by judging on the 20 most probable terms in each topic. If terms seem 1) semantically coherent in a way that a single topic label could be applied to them, and 2) this semantic coherence appears relevant for answering questions on democratic demarcation or self-conception in the discourse of the FRG, they appear as candidates for further processing. Although the rank_1 metric puts more specific topics in higher ranks of a listing, it still contains topics which are describing relatively general vocabulary. But the selection process also showed that interesting topics could be

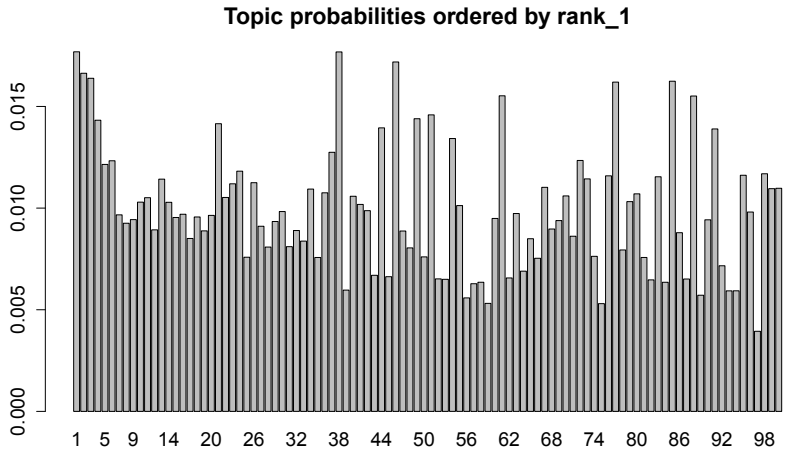


Figure 3.7.: Topic probabilities ordered by rank_1 metric. Higher bars throughout the entire rank range indicate that highly probable topics do not necessarily constitute primary document contents.

found mostly in higher ranks, while in lower ranks there were only few topic candidates for selection.

I selected the $K' = 10$ most meaningful topics for each cluster (see Table A.1). This selection also allows for a more formal evaluation of the two topic rankings just introduced. Topic rankings can be evaluated analogue to document ranking in IR (see Section 3.1.6). Computing Mean Average Precision (MAP) (Baeza-Yates and Ribeiro-Neto, 2011, p. 140) for topic ranking methods based on my manual selection assumed as 'gold standard' results in $\text{MAP}_{rank1} = 0.598$ for the rank_1 metric greatly outperforming $\text{MAP}_{prob} = 0.202$ for the ranking simply based on inferred topic probability.

For each of the K' manually selected topics per time period a SECG will be generated. This results in 50 SECGs for the exploratory analysis.

Topic Co-Occurrence

As documents are modeled as mixtures of topics, usually there are multiple topics in each document. As a first visual orientation towards the contents in each temporal cluster, we can identify co-occurrence of the manually selected topics in each cluster and visualize them as a graph network. For this, I count co-occurrence of the *two* highest ranked topics in each document using the rank_1 metric. This results in a symmetric $K' \times K'$ matrix M on which any significance measure for co-occurrence could be applied (Bordag, 2008). By using the LL metric and filtering for significant co-occurrences of any topic pair i, j above a threshold of $LL(i, j) >= 3.84$,[21] M may be transferred into a binary matrix M', where $M'_{i,j} = 1$ indicates a significant co-occurrence relation between topic i and topic j, and $M'_{i,j} = 0$ an insignificant relation. Then, M' can be used as an adjacency matrix for graph visualization. For each temporal cluster such a topic co-occurrence graph shows relevant topics and their relation (see Figure 3.8; graphs for the other four temporal clusters can be found in Chapter 4). Topics are displayed as nodes of the graph with the five most probable terms as their label. Node size indicates how often a topic has been inferred as primary topic within a document relative to the other topics in the graph.

3.2.5. Term Co-Occurrences

For constructing SECGs, we need to identify term co-occurrence patterns for each manually selected topic in each time period. The list of the most probable terms of a topic from the previously computed topic model provides a valuable basis for this. If combinations of terms co-occur significantly with each other in sentences of documents belonging to the selected topics, they are candidates for edges in the graphical visualization for corpus exploration. Co-occurrences are

[21] Rayson et al. (2004) describe this cut-off value for statistical LL tests for corpus linguistics corresponding to a significance level of $p < 0.05$.

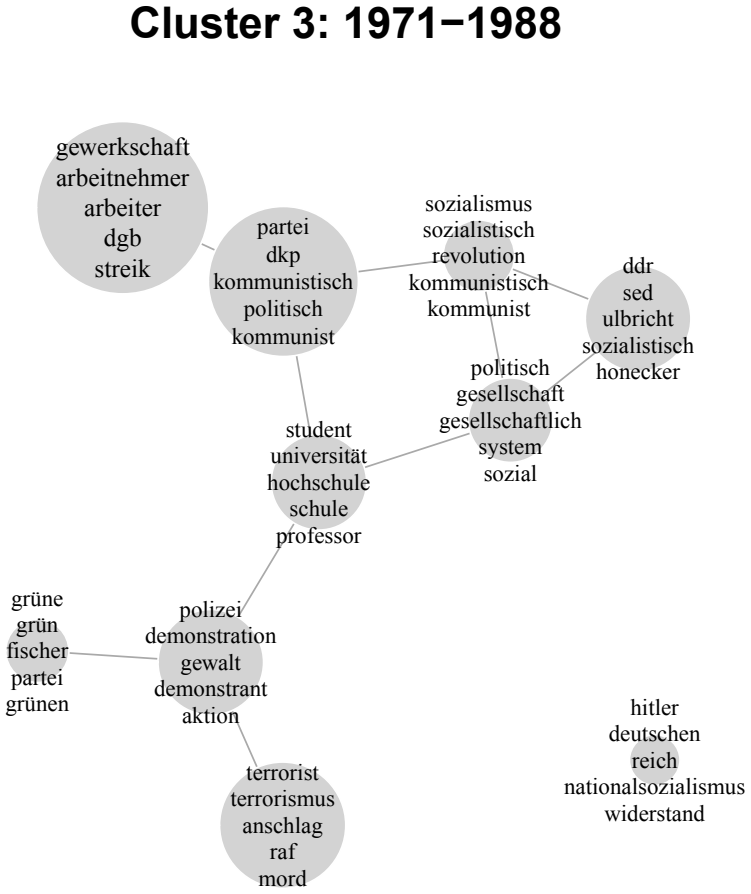


Figure 3.8.: Topic co-occurrence graph for 10 selected topics in cluster 3. Connected topics co-occur significantly as primary / secondary topic with each other in documents of that time period. Topic labels comprise of the five most probable topic terms.

extracted as follows (for a more formal description see Workflow 6 in the Appendix):

1. A set of documents $\mathcal{D'}_{c,k}$ of a time period c containing a topic share above a certain threshold $\theta_{d,k} > 0.1$ is selected. The topic mixture of the current topic model yields an average of 2.33 topics per document with a share greater than 0.1. This threshold ensures that only documents are selected, which contain topic k to a substantial share.

2. From the documents $\mathcal{D'}_{c,k}$ contained sentences $\mathcal{S'}_{c,k}$ are extracted and co-occurrence of the $N = 200$ most probable topic terms $V^k = (v_1^k, \cdots, v_N^k)$ within these sentences is counted.

3. Co-occurrence counts below a certain threshold $minC = 4$ are set to 0 to not focus on very infrequent or insignificant events.

4. Significance of co-occurrence counts $sig(a, b)$ for two terms $a \in V^k$ and $b \in V^k$ is computed using the LL measure (Bordag, 2008, p. 54f) with respect to the size of the entire sentence set $n = |\mathcal{S'}_{c,k}|$.

$$\lambda = \left[\begin{array}{l} n \log n - n_a \log n_a - n_b \log n_b + n_{ab} \log n_{ab} \\ +(n - n_a - n_b + n_{ab}) \log(n - n_a - n_b + n_{ab}) \\ +(n_a - n_{ab}) \log(n_a - n_{ab}) + (n_b - n_{ab}) \log(n_b - n_{ab}) \\ -(n - n_a) \log(n - n_a) - (n - n_b) \log(n - n_b) \end{array} \right]$$

$$sig(a, b) = \begin{cases} -2 \log \lambda, & n_{ab} < \frac{n_a n_b}{n} \\ 2 \log \lambda, & \text{otherwise} \end{cases} \tag{3.21}$$

   where n_a and n_b are the number of sentences in $\mathcal{S'}_{c,k}$ containing a, or b respectively; n_{ab} is the number of sentences containing both terms.

5. Significance values below a certain threshold $minLL = 3.84$ are set to 0 to not focus on insignificant term relations.[22]

[22] For the significance thresholds, see footnote 21.

Table 3.9.: Examples of extracted co-occurrences per temporal and thematic cluster.

Term 1	Term 2	LL
Cluster 2: Topic 59		
diplomatisch	beziehung	934.745
kalten	krieg	745.370
teil	deutschlands	576.233
west	ost	507.562
anerkennung	ddr	502.969
kalt	krieg	438.752
bundesrepublik	ddr	372.925
osteuropäisch	staaten	319.468
stellen	frage	314.774
völkerrechtlichen	anerkennung	306.561
Cluster 3: Topic 62		
dienst	öffentlich	2609.087
grundordnung	demokratisch	610.448
eintreten	grundordnung	571.928
jederzeit	eintreten	514.733
verfassungsfeindlich	partei	481.145
freiheitlichen	grundordnung	479.399
jederzeit	grundordnung	418.212
freiheitlichen	demokratisch	413.246
mitgliedschaft	partei	401.283
verfassungsfeindlich	mitgliedschaft	383.501

6. Pairs of significant co-occurrences are ordered decreasingly to their LL-value.

Extracted co-occurrence pairs are the basis for the visualization of SECGs. An example for the top 10 co-occurrence pairs extracted from two topics is given in Table 3.9. In the following sections, enrichment of additional semantically insightful data on single terms participating in co-occurrence relations is described.

3.2.6. Keyness of Terms

Not all terms taking part in a significant co-occurrence relation are equally descriptive for the contents of the topic. To judge on relative importance of terms, we need to apply a statistical measure. One possible measure would be the term probability given the topic $P(t|\beta_k)$. But we do not want to weight terms globally based on the entire corpus. Instead, we want to base the judgments with respect to the topic and time period under investigation. The simplest idea would be to apply frequency counts of terms in these sub-corpora to determine their importance. But we already know, frequency alone is a bad indicator for keyness. Better measures are those which are established for key term extraction.

In Section 3.1.2, the Log-likelihood (LL) measure is described for key term extraction. As the results for document retrieval indicate its usefulness, we simply employ it once more to judge on relevancy of the terms taking part in our extracted co-occurrence relations. As a comparison collection for computing the LL measure, again the corpus $\mathcal{W}$ compiled from 100,000 randomly selected sentences from Wikipedia articles is taken (see Section 3.1.2). For every term of the $N = 200$ most probable topic terms in each SECG relative overuse to this Wikipedia corpus is computed. Table 3.10 displays examples of term keyness for two topics in two distinct time clusters..

3.2.7. Sentiments of Key Terms

Sentiments[23] expressed towards certain entities can be a valuable heuristic device to explore large document collections. Entities could be terms, concepts (lists of terms), term co-occurrence pairs, single documents or even entire topics. As terms in co-occurrence graphs can be taken as representatives of discursive signifiers within a thematic coherence, observation of sentiment terms expressed within their contexts might be a valid approach to reveal emotions to these entities in the discourse. Especially for investigating speech acts on democratic

[23]See Section 2.2.3 for some introductory notes on Sentiment Analysis in QDA.

Table 3.10.: Example key terms extracted per temporal cluster and topic.

Cluster 2: Topic 59		Cluster 3: Topic 62	
Term	**LL**	**Term**	**LL**
deutschen	17179.936	partei	7683.841
bundesrepublik	14244.400	dkp	6531.410
ddr	9833.421	öffentlich	4483.212
politisch	9745.186	dienst	4460.154
politik	8529.275	bewerber	3944.922
deutschland	6839.388	beamte	3803.593
wiedervereinigung	6216.722	kommunistisch	3201.241
frage	5716.529	grundgesetz	2913.200
bonn	5535.869	verfassungsfeindlich	2867.454
deutschlands	5199.870	demokratisch	2660.121

demarcation, one would expect normative or moral language towards certain actors, ideas or activities to be found in news coverage.

To enrich co-occurrence graphs with sentiment information, I compute a sentiment score for each term it consists of. For this, a rather general basic approach is used. The selected approach has the advantage of easy implementation and also allows for comparability of the results aggregated on topic level to a certain degree. For detecting sentiments, I employ the German dictionary resource SentiWS (Remus et al., 2010). SentiWS provides two lists of weighted lemmas together with their inflected forms—one list of 1,650 positive terms and one list of 1,818 negative terms. Besides category information on polarity, each term t in SentiWS is assigned with a polarity weight w_t. Positive terms are weighted on a scale between [0;1]; negative terms are weighted on a scale between [-1;0] (see examples in Table 3.11). The lists have been compiled by an automatic process which initially relies on a seed set of definitely positive or negative words (e.g. *gut, schön, richtig, ...* or *schlecht, unschön, falsch, ...*). Polarity weighting for other terms (named target terms in the following) is then performed by observing co-occurrence between these target terms and the either positive or negative seed terms in sentences of an example corpus. Co-occurrence

Table 3.11.: Examples of positive and negative terms together with their polarity weight w_t in the SentiWS dictionary (Remus et al., 2009).

Positive		**Negative**	
Term	$\mathbf{w_t}$	**Term**	$\mathbf{w_t}$
Aktivität	0.0040	Abbau	-0.058
Befreiung	0.0040	Bankrott	-0.0048
Leichtigkeit	0.1725	Belastung	-0.3711
Respekt	0.0040	Degradierung	-0.3137
Stolz	0.0797	Lüge	-0.5
beachtlich	0.0040	Niederlage	-0.3651
bewundernswert	0.0823	aggressiv	-0.4484
hochkarätig	0.0040	alarmieren	-0.0048
knuddelig	0.2086	furchtbar	-0.3042
toll	0.5066	gewaltsam	-0.0048

counts for target terms with seed list terms are judged for statistical relevance by the Pointwise Mutual Information (PMI) measure and finally aggregated to a score on semantic orientation. The approach is based on the assumption that terms of a certain semantic orientation co-occur more frequently with terms of the same orientation than of the opposite. Evaluation of this approach with human raters shows that the performance of identifying positive/negative terms correctly is "very promising ($P = 0.96, R = 0.74, F = 0.84$)" (Remus et al., 2009, p. 1170).

To infer on sentiments of each term of the N most probable topic terms $V^k = (v_1^k, \cdots, v_N^k)$ for a topic k in a specific time period, I apply the SentiWS dictionary in the following manner:

1. Analogue to extraction of co-occurrences (see Section 3.2.5), for each time cluster c a set of documents ${\mathcal{D}'}_{c,k}$ containing a share of topic k above a threshold $\theta_{d,k} > 0.1$ is identified.

2. For each term $v_i^k \in V^k$
   a) Extract a set $\mathcal{S}_i$ of sentences from ${\mathcal{D}'}_{c,k}$ which contain v_i^k

b) Count frequencies $\mathbf{n}$ of sentiment terms $t \in$ SentiWS: Set $n_t \leftarrow tf(t, \mathcal{S}_i)$, if $tf(t, \mathcal{S}_i) > 0$

c) Multiply all sentiment term frequencies $\mathbf{n}$ with their respective polarity weight $\mathbf{w}$ from SentiWS: $s_t \leftarrow w_t n_t$

d) Compute a sentiment score p_i for v_i^k by averaging over all polarity weighted SentiWS term counts $\mathbf{s}$: $p_i \leftarrow \bar{s} = \frac{1}{|\mathbf{s}|} \sum_{j=1}^{|\mathbf{s}|} s_j$

e) Compute a controversy score q_i for v_i^k by determining the variance of all polarity weighted SentiWS term counts: $q_i \leftarrow var(\mathbf{s}) = \frac{1}{|\mathbf{s}|-1} \sum_{j=1}^{|\mathbf{s}|} (s_j - \bar{s})^2$

3. An overall sentiment score $P_{c,k}$ for the entire topic in that time frame can be computed by summing up all sentiment scores $\mathbf{p}$: $P_{c,k} = \sum_{i=1}^{N} p_i$

4. An overall controversy score $Q_{c,k}$ for the entire topic in that time frame can be computed by taking the variance of all sentiment scores $\mathbf{p}$: $Q_{c,k} = \frac{1}{n-1} \sum_{i=1}^{N} (p_i - \bar{p})^2$ where $\bar{p}$ is the mean of $\mathbf{p}$.

This procedure provides a sentiment score and a controversy score for each term in one SECG. Furthermore, by computing variances of sentiment scores per term and topic, we may identify terms / topics which are highly debated. This may be assumed because we observe a broader range of positive and negative contexts for the most probable terms of a topic. Table 3.12 gives examples for highly positive and negative as well as (non-)controversial terms identified in two topics of two time periods.

3.2.8. Semantically Enriched Co-Occurrence Graphs

After having extracted various information from our to-be-explored corpus $\mathcal{D}'$ of 28,557 documents, we can now put it all together to visualize Semantically Enriched Co-occurrence Graphs (SECGs):

For each sub-collection $\mathcal{D}'_{c,k}$ selected by temporal cluster c (see Section 3.2.3) and topic k (see Section 3.2.4), we combine the extracted information as follows:

Table 3.12.: Examples of sentiment (p_i) and controversy scores (q_i) for terms per temporal cluster and topic.

Cluster 2: Topic 59		**Cluster 3: Topic 62**	
Term	**p_i**	**Term**	**p_i**
erfolg	0.1259	aktiv	0.0094
menschlich	0.1035	angestellte	0.0090
völkerrechtlichen	0.0321	gewähr	0.0029
erleichterung	0.0287	jederzeit	-0.0038
normalisierung	0.0268	sinn	-0.0067
anerkennung	0.0259	anfrage	-0.0085
drüben	0.0248	ziel	-0.0119
entspannung	0.0209	beamte	-0.0148
...	...	...	...
überwindung	-0.0141	extremist	-0.0482
recht	-0.0141	rechtfertigen	-0.0505
mauer	-0.0147	pflicht	-0.0592
offen	-0.0157	streitbar	-0.0598
teilung	-0.0169	absatz	-0.0634
endgültig	-0.0213	zugehörigkeit	-0.0743
verzicht	-0.0352	verboten	-0.0866
kalt	-0.0698	ablehnung	-0.1068
Term	**q_i**	**Term**	**q_i**
krieg	3.2601	radikal	0.7370
anerkennung	1.4999	verbieten	0.4616
erfolg	1.2228	ablehnung	0.3971
deutschen	0.6103	partei	0.2410
menschlich	0.5139	öffentlich	0.1485
bundesrepublik	0.4922	verboten	0.1358
politisch	0.3737	dienst	0.1301
politik	0.3625	pflicht	0.0806
...	...	...	...
wiederherstellung	0.0022	bundesverwaltungsgericht	0.0010
west	0.0021	gewähr	0.0008
status	0.0019	prüfen	0.0007
friedensvertrag	0.0018	absatz	0.0006
hallstein-doktrin	0.0017	frankfurter	0.0006
überwinden	0.0017	angestellte	0.0006
normalisierung	0.0015	einzelfall	0.0006
wiedervereinigen	0.0014	freiheitlich-demokratische	0.0005

1. **Graph:** Construct a co-occurrence graph $G = (V, E)$ based on extracted co-occurrence pairs (see Section 3.2.5). Vertices V are defined by terms taking part in the j most significant co-occurrence relations. Edges E are defined by existence of the co-occurrence pair relation between two terms of V. To keep the visualization clear, I set $j = 60$. This leaves aside many significant co-occurrence patterns, but helps to concentrate on the most important ones. Furthermore, disconnected sub-graphs which contain less than 3 nodes are removed from G, to not inflate the graph with many isolated term pairs. Usually, those isolated term pairs represent typical collocation patterns of general language regularities of the German language rather than specific thematic content. Removing them, puts emphasis on the giant component of the graph.

2. **Edge size:** Edges E of G are weighted by LL-significance of their co-occurrence. For visualization, the more significant the term relation, the thicker an edge will be drawn.

3. **Vertex size:** Vertices V of G are weighted by 'keyness' of their occurrence (see Section 3.2.6). For visualization, the higher the LL score of a term, the bigger the Vertex will be drawn. Vertices are labeled with the representing terms. Label sizes are also scaled along with vertex sizes to emphasize on 'keyness'.

4. **Vertex color:** Vertices V of G are colored according to their contextual sentiment (see Section 3.2.7). Vertices representing negative terms will be colored in a range from *red* $\widehat{=}$ most negative term to *grey* $\widehat{=}$ no sentiment. Vertices representing positive terms will be colored in a range from *grey* $\widehat{=}$ no sentiment to *green* $\widehat{=}$ most positive term. To translate sentiment scores into color palette values, scores are re-scaled into a value range of [0; 1].

5. **Vertex frame color:** Frames of vertices V are colored according to their controversy score (see Section 3.2.7). Controvery scores will be translated into a color range from *white* $\widehat{=}$ low controversy score to *orange* $\widehat{=}$ high controversy score. For selecting suitable color palette values, scores are re-scaled into a value range of [0; 1].

6. **Edge color:** The structure of G can be utilized to heuristically identify semantic propositions, i.e. assertions on entities within sentences. For this, one can identify maximal cliques in G of size 3 or higher. Vertex sets of these cliques represent fully connected sub-graphs of G which means that all participating terms co-occur significantly with each other in sentences of a topic and time period. These patterns reveal important meaningful language regularities, constituting central discursive assertions. Edges which are part of such a maximal clique are colored *green*. Due to vertex removal in step 1, it may happen that the clique structure is not represented any longer in G. Consequently, maximum cliques of size $>= 3$ should be identified before vertex removal such that all previously extracted co-occurrence pairs are used. Table 3.13 gives examples for extracted propositional candidates.

7. **Text examples:** In addition to global contexts represented by co-occurrence graphs, qualitative information for each SECG is provided by extracting ranked lists of 'good' text examples. For this, candidates for semantic propositions from the previous step are utilized to select sentences from $\mathcal{D}'_{c,k}$ containing all of its components. For each proposition candidate, I sampled five example sentences from $\mathcal{D}'_{c,k}$. Sets of sampled sentences are ranked according to summed LL-significance values of the co-occurrence relation it consists of.

For five temporal clusters, each with 10 manually selected topics, we can draw a SECG.[24] We get 50 SECGs supporting the content analyst by getting a quick visual overview of the most important topics during different time frames. The final SECG for the two topics/periods used throughout this section are given in Figures 3.9 and 3.10.

Further, providing good text examples together with each SECG allows for qualitative assessment of the extracted global contexts which

[24] I utilized the *igraph* package (Csardi and Nepusz, 2006) for R to generate the Graphs. Vertices are arranged on the canvas using the Fruchterman-Rheingold layout algorithm.

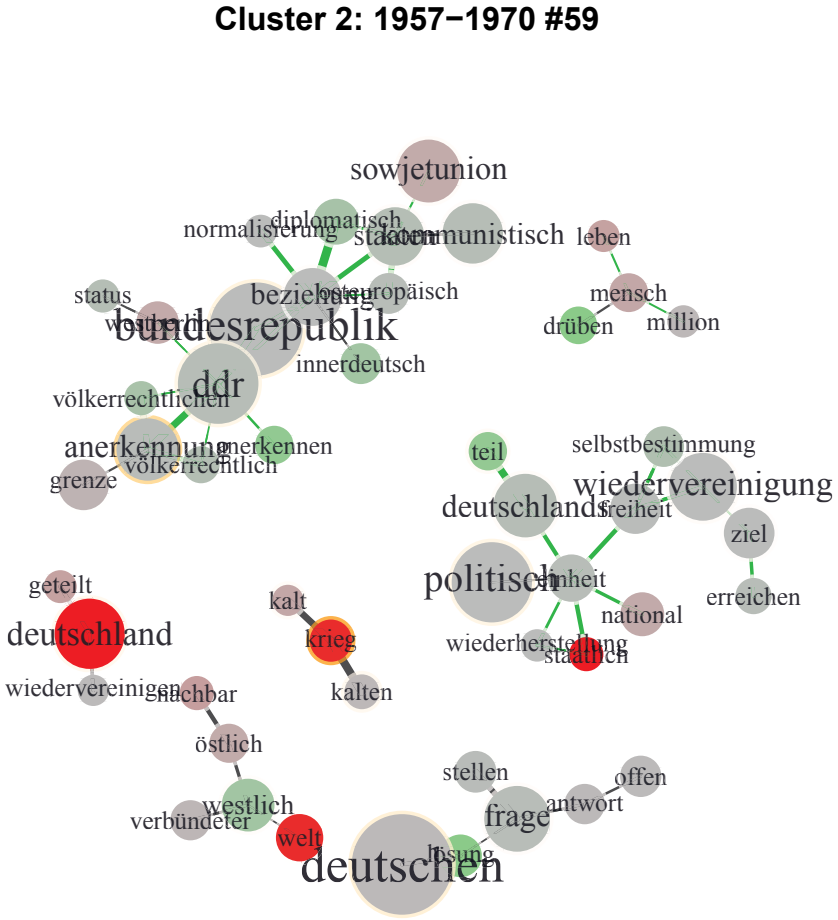


Figure 3.9.: Example of SECG (cluster 2, topic #59).

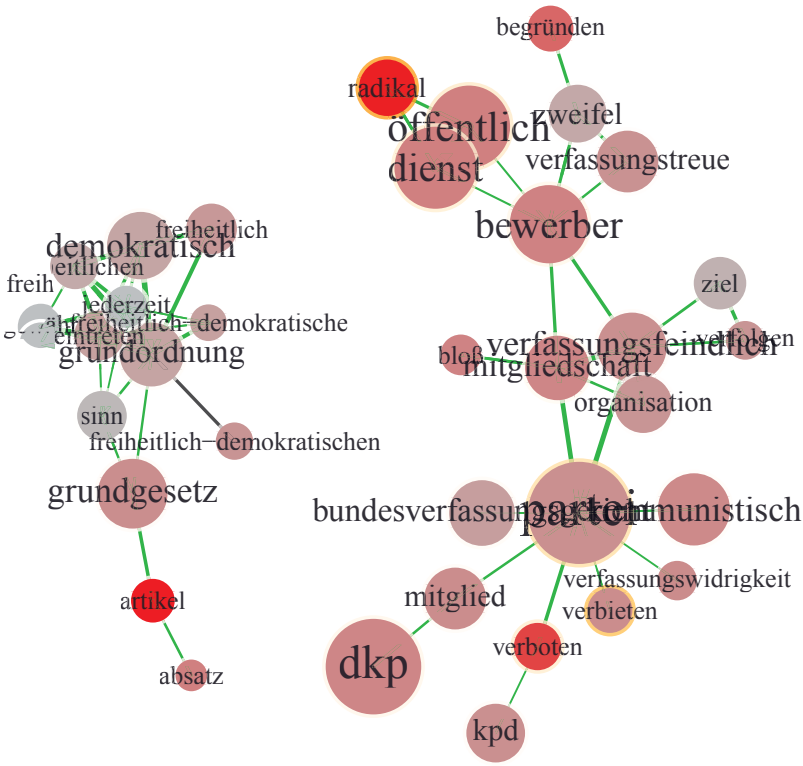


Figure 3.10.: Example of SECG (cluster 3, topic #62).

Table 3.13.: Examples for candidates of semantic propositions identified from maximum cliques in SECGs.

Cluster 2: Topic 59	**Cluster 3: Topic 62**
normalisierung, beziehung, ddr	überprüfung, erkenntnis, bedenken
wiedervereinigen, deutschlands, status	verfassungsfeinde, öffentlich, dienst
leben, mensch, million	verfassungswidrig, partei, erklären
westdeutsch, bundesrepublik, ddr	absatz, grundgesetz, artikel
hallstein-doktrin, beziehung, diplomatisch	zweifel, bewerber, begründen, jederzeit, eintreten, grundordnung
erreichen, wiedervereinigung, ziel	zugehörigkeit, partei, verfassungsfeindlich
existenz, ddr, anerkennen	extremist, öffentlich, dienst
teilung, europas, teilung deutschlands	ordnung, demokratisch, freiheitlich
anspruch, selbstbestimmung, wiedervereinigung	feststellen, bundesverfassungsgericht, verfassungswidrigkeit
teilung, europas, überwindung	kpd, dkp, verboten

is an important feature to support QDA. Table 3.14 gives examples for extracted sentences allowing for much better interpretation of semantic relational structures visualized by SECGs. In fact, text examples are selected by a notable back and forth mechanism. The data-driven process of generating SECGs reveals linguistic patterns on the global context level within a certain topic and time frame. Using such globally identified patterns to look for local contexts comprising of all of its features allows for selection of high quality examples incorporating central semantics of the topic. As vertices in G represent binary relations of co-occurrence, it is not guaranteed to find sentences or propositions containing all three or more components. But usually, at least some sentences can be retrieved, which then can be interpreted as good candidates containing sedimented discourse characteristics.

Table 3.14.: Examples of text instances containing previously identified semantic propositions (see Table 3.13).

Cluster 2: Topic 59	**Cluster 3: Topic 62**
"Es sei daher der Abschluß eines Vertrags zwischen allen Staaten Europas über den Gewaltverzicht nötig, ebenso 'die Normalisierung der Beziehungen zwischen allen Staaten und der DDR wie auch zwischen den beiden deutschen Staaten und zwischen West-Berlin (als besonderem politischem Raum) und der DDR'".	"Der umstrittene Extremistenbeschluß der Länderministerpräsidenten vom Januar 1972 setzt für Bewerber um ein Staatsamt weit strengere Maßstäbe: 'Gehört ein Bewerber einer Organisation an, die verfassungsfeindliche Ziele verfolgt, so begründet diese Mitgliedschaft Zweifel daran, ob er jederzeit für die freiheitliche demokratische Grundordnung eintreten wird.'"
"In der ruhigen, unablässigen Forderung der Freiheit der Selbstbestimmung, ohne Verknüpfung mit dem Anspruch der Wiedervereinigung, haben wir die Unterstützung durch unsere Verbündeten und der Weltmeinung kräftiger, rückhaltloser, eindeutiger für uns."	"'Verfassungsfeinde gehören nicht in den öffentlichen Dienst,' bekräftigte der Minister Anfang April im CSU-Organ Bayernkurier noch einmal."

3.2.9. Summary of Lessons Learned

The introduced process of generating SECGs provides an intuitive access for content analysts to explore large data collections. Knowledge structures inherent to the collection are visualized on a global context level suitable for 'distant reading'. Text snippets with high informative value based on extracted semantic structure are provided with each graph to backup interpretations from visualized display by qualitative data review. Based on the requirements initially formulated, the following insights have been obtained during this section:

- Document collections can be separated, both temporally and thematically, into small, coherent segments by using topic models. Optimal parameters for topic modeling can be obtained with the topic coherence measure (Mimno et al., 2011) alongside with qualitative

evaluation of the topic results. For quality assurance, reproducibility of the model can be measured. Temporal segmentation of the data into distinctive time periods can be achieved in a data-driven manner by PAM clustering on the inferred topic proportions.

- Selection of meaningful topics from a topic model with respect to a specific research question should be done deliberately by the researcher in a manual process. Nonetheless, it can be supported by ranking topics, e.g. according to the rank_1 measure, i.e. the number of their primary occurrence in documents.

- Significant co-occurrence of topics in temporally segmented sub-collections can be visualized as network graph to reveal global thematic structures.

- Significant co-occurrence of terms in temporally and thematically segmented sub-collections can be visualized as network graph to reveal patterns of language use for content exploration. Term co-occurrence networks can be enriched by additional semantic information, such as sentiment and keyness of terms in their thematic contexts, visualized by color or size of graph vertices.

- Graph structures in co-ocurrence graphs such as maximal cliques reveal semantic fields of paradigmatically related terms which can be assumed as candidates for semantic propositions. Candidates for propositions point analysts to potentially interesting categories[25] for further investigations (see Section 3.3). Text snippets such as sentences containing these semantic propositions appear to be excellent data samples to backup interpretations from the global contexts of graphs qualitatively.

The generation of SECGs is not an entirely unsupervised process. While most parts are purely data-driven, analysts still need to decide for specific parameters at certain points. This should not be seen as a

[25] I use 'categories' here in the sense of language regularities constantly re-used over time which may give hints to sedimented discourse structures.

weakness, rather than a strength of the process. It provides analysts with opportunities to keep control over the analysis and to check for validity and compliance with their background knowledge. Retaining control in such exploratory workflows is a necessary precondition to develop confidence in the computationally produced results. To make things easier to apply, algorithms and quality measures may provide hints for best choices of parameters. But in the end, it should be the researchers decision to select

- an appropriate number of topics K for the topic model to achieve desired thematic granularity,
- a plausible α value for regulating topic distributions in documents,
- a comprehensible number of time periods, and
- a manageable number and conscious (manual) selection of topics per time period to draw graphs from.

While analysis capabilities and quality of results truly increase, given the analyst understands fundamentals of these steps, profound understanding of algorithmic details is not needed necessarily to produce useful results. Sticking to default values and data-driven optimal parameter suggestions will also lead to valuable results in most cases.

The method presented here is an exemplary application which provides a strategy tailored to the research needs and requirements for exploring the data set on democratic demarcation. Further modifications to this process could be made based on different operationalization decisions, e.g. using some other topic model instead of LDA, altering the way of how the graph vertices and edges are defined, including other text statistical measures into the visualization or choosing another layout algorithm for the graph.

As this section focused mainly on the technical realization of the presented workflow, open questions remain on the methodological aspects. From QDA perspective, researchers need to learn to integrate results from such processes into their general analysis. They need to describe steps they take comprehensibly and in a manner allowing

for reproduction. Furthermore, they need to perform careful and theoretically sound interpretation of results in the light of their methodological background. Some thoughts in this direction are elaborated on in Chapter 5.

3.3. Classification for Qualitative Data Analysis

In QDA, methods of coding text operate either inductively, deductively or as a mix of both approaches, also sometimes referred to as abductive paradigm. Inductive research develops its categories from observations in the empirical data and can be supported by exploratory tools, as presented in the previous chapter. For the deductive approach, usually categories of content are derived from text external theoretical assumptions. Abductive research develops its categories from (samples of) the data and, afterwards, utilizes category systems for subsuming new data and hypothesis testing (Kelle, 1997). To support subsumptive coding of text as essential part of a QDA process, we will augment our exploratory analysis conducted in the previous section (see 3.2) by identifying concrete categorical content in the data concerned with several aspects of democratic demarcation. To prepare this step, I compose a workflow of CA utilizing supervised ML to extend manual analysis capabilities to large amounts of textual entities. The process addresses specific requirements of the social science domain and will be evaluated with example data to determine its usefulness for time series and trend analysis.

In manually conducted CA, trained persons, called coders, categorize textual entities by hand. They read through quantities of material, either the full data set under investigation or a sample randomly drawn from it, and attach a code label, if a certain entity fits into the definition of a category. Categories are defined together with example snippets in so called “code books” which try to describe a category as accurately as possible (Krippendorff, 2013). Textual entities under investigation might be from varying granularity: words, phrases, sentences, paragraphs or whole documents. In most QDA applications

researchers are interested in 'propositions'—certain meaning expressed in a declarative sentence or sequence of sentences.[26] Much of this manual coding effort can be supported by computer-assisted methods to a certain extent. The most promising innovation in these analytic procedures can be expected from supervised machine learning, also referred to as classification.

Classification of text has been a broad research area in NLP for several decades. Applications range from email spam detection and genre identification to sentiment analysis. Formally, we can define the classification problem as a binary function on a set of documents $\mathcal{D}$ and a set of classes $\mathcal{C}$ in the following manner: the function $\mathcal{F} : \mathcal{D} \times \mathcal{C} \rightarrow \{0, 1\}$ assigns either 0 or 1 to a pair $[d_j, c_p]$ where $d_j \in \mathcal{D}$ and $c_p \in \mathcal{C}$. An assigned value 0 indicates that d_j does not belong to class c_p, 1 indicates it does belong to c_p (Baeza-Yates and Ribeiro-Neto, 2011, p. 283). A classification algorithm provides such a function which strives to fulfill this assignment as accurately as possible with respect to the empirical data. For this, it extracts characteristic patterns, so called features, from documents of each category in a training phase. It therewith 'learns' these pattern–class associations to build the function $\mathcal{F}$, which also may be called an instance of a classification model. With this model instance, it now is possible to assign class labels to unknown documents by observing their feature structure. Concerning the fact that the model based on training data is necessarily incomplete with regard to all existing data in a population, prediction cannot be fully exact. The quality of a model instance can be evaluated by well established quality measures such as *accuracy*, *precision*, *recall* and F_1 (Asch, 2013; Baeza-Yates and Ribeiro-Neto, 2011, p. 325) which will also be utilized throughout this study.

Although supervised text classification already has a long history in NLP, it has not been applied in QDA widely. In NLP investigation of problems in text classification usually is done by using standard

[26] For example, Teubert (2008) investigates political positions towards the European Union (EU) expressed in British Online Forums. Wiedemann (2011) investigates German parliamentary debates to identify argumentative patterns for or against data retention in telecommunication.

corpora like Reuters-21578 Text Categorization Collection (newswire texts from 1987), 20 Newsgroups data set (news group articles on sports, computers etc.) or abstracts of (bio-)medical research papers. Classes of these corpora usually are genre or topic related and rather clearly defined; classification experiments are based on label assignments to complete documents rather than snippets of documents. In real world applications of QDA such "laboratory conditions" unfortunately are seldom met. A first systematic study on applicability of ML classifiers for CA in the German context has been conducted by Scharkow (2012). Although this study states the usefulness of fully automated analysis, it also operates on rather simple genre categories of newspaper data (e.g. identifying newspaper sections such as sports, politics or culture).

Applying supervised machine learning in QDA scenarios is a challenging task. The purpose of this section is to provide a workflow to employ this technology as effective as possible within a QDA process. For this, I firstly describe requirements of that analysis task which differ from standard NLP classification scenarios in several ways. Then, I conduct experiments on real world data from a political science project on hand coded party manifestos. Base line classification accuracy of propositional categories is evaluated and compared to an extended feature set incorporating topic model data as features for 'semantic smoothing' of the data. In a third step, applicability of classification for trend identification is demonstrated. In a last step, I propose an active learning workflow to create training data for classification with low cost and high quality for the desired purpose. Thus, this section answers the questions:

- How good can automatic classification for QDA purposes be?,
- How exact has automatic classification to be to produce valid results for trend analysis? and,
- How can training data be collected effectively by active learning?

Experimental setups and the resulting best practice for applying ML in QDA scenarios are employed in the subsequent Chapter 4. The goal

is to identify and investigate propositional categories on democratic demarcation in the collection $\mathcal{D}$ comprising of all *FAZ* and *Die Zeit* articles.

3.3.1. Requirements

Manual CA conducts studies on randomly drawn samples of text from certain, well-defined populations to infer on category distributions or proportions within these populations. This works well, as long as there are lots of manual coders and the number of populations where samples are drawn from is fixed and small. To investigate a category, e.g. statements expressing democratic demarcation towards (neo-)fascist ideology, in the 1950s, it would be acceptable to draw a representative random sample, hand code its content and measure code proportions. But it certainly would not be justifiable to infer on proportions in subsets of that basic population. For example, to infer on category proportions in 1951 compared to 1952 or ongoing years, we probably neither have enough hand coded data, nor do we have representative samples. To compare proportions for a time series, we would need to draw random samples of sufficient size from each time frame and manually code them. In this scenario clear advantages of (semi)automatic classification procedures come into play. A well trained classification model allows for reliable measurement of categories in varying subsets of its base population. This is because it predicts on each individual case of the entire base population whereas each case is classified independently of each other. But how reliable can machine classification be in contrast to (well-trained) human coders under circumstances of QDA?

Text classification for QDA faces several distinctive challenges in contrast to standard text classification scenarios in NLP which need to be addressed, if supervised machine learning should be applied successfully to an analysis workflow:

- **Abstract categories:** Categories of interest in QDA often are much more abstract than simple news genre labels like *sports*,

politics or *weather*. Textual units representing desired categories are expressed in a potentially unlimited, heterogeneous variance of word sequences. In practice, the overall majority of expressions constituting a certain category is formed by only a small number of variants. Human discourse tends to use similar expressions to express similar things, which yields regular patterns in language use. This is why machine classification (as well as human understanding) of texts can be successful in the first place—the identification of patterns in language use and their mapping to categorial semantic units. Nonetheless, categories relevant in QDA studies, such as expression of democratic demarcation, economized argumentation in politics (Wiedemann et al., 2013) or ethnicized reporting in news coverage (Pollak et al., 2011) are not only identifiable by certain key words alone. In case of simple categories, the observation of the term *soccer* might be a decent indicator for a general genre category *sports*. Most QDA categories instead are constituted by complex combinations of sets of terms and even syntactic structure. Thus, employed classification algorithms should be capable of taking many different features as well as dependence of features into account.

- **Unbalanced classes:** While classes in standard evaluation corpora are mostly of comparable size[27], classes in CA contexts are highly unbalanced. Imagine again the measurement of statements against (neo-)fascist attitudes: even in a newspaper article dealing with current activities of the far right there are probably only a handful out of thirty to forty sentences expressing "demarcation" in the sense of the desired category. Odds between positive and negative examples for a desired context unit may be 1:20, 1:50 or 1:100. Classification algorithms need to be able to deal with these discrepancies.

- **Sparse training data:** Text classification for entire documents is the standard case in many NLP applications. As algorithms usually

[27] The Reuters-21578 corpus contains some very low frequent classes as well, but most studies leave them out for their experiments.

are based on vector representations of the units to be classified, document classification has a clear advantage over classification of smaller units such as paragraphs or sentences which are the 'natural' units of many CA applications. Vectors representing such units of text are much more sparse than vectors of complete documents, putting less information on features into the classification process. To address this problem, we need to engineer features representing more generalized context than just the few words contained in a single sentence or paragraph. We should also try not to 'learn' from the hand coded training data only, but in a semi-supervised classification scenario from the freely available unlabeled data of our to-be-classified corpus as well.

- **Small training data:** Standard evaluation procedures in NLP deal with scenarios where training data is abundant. In contrast to this, QDA studies investigate categories fitting a special research interest. Unfortunately, manual coding of examples is labor intense and therefore costly. For this reason, QDA studies are restricted to a limited number of training data they can generate. This situation poses different questions: Which classifier should be taken? Some classification algorithms are able to deal with small training data better than others. Can the process of generating training data be optimized by application of active learning, to get best possible results at low costs?

- **Social science goals:** Classification in the standard case tries to optimize accuracy of individual prediction for individual cases. This definitely makes sense for applications such as spam detection on emails, where we want to avoid false positives, i.e. emails deleted as spam although they are not spam. For QDA studies on large data sets, we are not so much interested in evaluating each individual case. We merely are interested in estimating proportions of categories in populations correctly (Hopkins and King, 2010). Even less restrictive, we might be interested in observing trends in the quantitative development of categories over time. In this case, even category proportions would not need to be overly exact,

as long as the estimation errors for proportions in time slices of the basic population are stable. To determine the usefulness of machine classification for CA, we need to clarify at first how well it can perform with respect to conventional evaluation procedures in principle. We can expect a lowered performance compared to acceptable results of standard NLP tasks, because of the hard conditions of this task. Nonetheless, if we modify the evaluation criteria from correct individual classification towards the goal of observing proportions and trends validly and reliably, we might be able to prove the usefulness of the method for trend and time series analysis.

The following sections address these requirements and formulate practical solutions to optimize machine classification of QDA categories with respect to social science goals. For this, experiments on real world data are conducted to determine reliability and validity of the overall approach, as well as identifying best practices. Results are also compared to a method of "proportional classification", suggested by Hopkins and King (2010), which addresses some of the requirements introduced above.

Category Systems

In supervised classification three types are usually distinguished:

1. single-class: decision whether or not an item belongs into a single category (e.g. spam vs. no spam),

2. multi-class: decision to assign exactly one class label to an item from a set of three or more classes,

3. multi-label: decision whether an item belongs into one or more classes from a set of three or more classes.

The third case can be treated as a repeated application of the first case with each label separately. Hopkins and King (2010) propose a fourth type of 'proportional classification' which is not interested in labeling

individual items, but in estimating correct proportions of categories within a population under investigation. For QDA purposes in social sciences, all four types of classification might be valid approaches in certain scenarios. But usually, the nature of categories of interest in QDA studies is related to the single-class / multi-label case. To clarify this, we first look at the multi-class variant more closely. Multi-class scenarios require category systems with two decisive properties:

- completeness: the set of categories needs to describe each case of the population, i.e. one label needs to be applicable meaningfully to any item,
- disjointness: categories should not be overlapping, i.e. that exactly one label of the set of categories should apply to any item of the population.

For most category systems applied in QDA studies, these conditions are not met. The property of completeness might be mitigated by introducing a special category *'irrelevant item'* to the code book which could be used to label all items which do not contain meaningful content for the study. More complex is the problem of disjointness. Categories in many QDA applications are not clearly separable. In fact, overlapping of categories in concrete items of empirical data might be of special interest for observation. These cases may represent co-occurrence of ideas, thoughts, discourses, and, hence, indicate certain argumentative strategies. Category systems could also be hierarchical, where sub-categories cannot be defined as disjoint cases, but as different perspectives on the root category which can occur conjointly in single items. For this reason, I suggest to concentrate on the single-class case for studying the applicability of machine classification for QDA. The multi-label case is treated as n cases of the single-class classification.

3.3.2. Experimental Data

In the following, several experiments are conducted to derive a reasonable workflow for the application of machine classification in QDA.

Final goal of this workflow is to infer on category proportion development over time to describe certain aspects in the discourse on democratic demarcation in Germany. But as we do not know anything about these categories yet, we have to refer to another experimental data set, to develop and evaluate an analysis strategy.

For the experiments, I rely on extracts of the data set of the *Manifesto Project Database*[28]. The Manifesto Project (MP) collects party manifestos from elections worldwide and conducts manual CA on them (Volkens et al., 2014). Each sentence of a single manifesto is annotated by trained coders with one (in rare cases also more than one) of 57 categories. Categories comprise of demands towards certain policy issues such as economics, welfare state, environment or foreign politics. The database contains frequencies of categories for each manifesto per party and election. Political scientists use this data to quantitatively compare distributions of policy issues over various dimensions (e.g. time, party, country, political spectrum).[29] It thus provides an excellent resource of high-quality 'real world' text data, which also can be used for experiments with machine classification.

For experimentation with MP data, I selected all available hand coded full-text party manifestos of the major German parties from the last elections (see Table 3.16). The total data set comprises of 44,513 manually coded sentences. To be coherent with my topic on democratic demarcation, I selected four categories out of the MP category set, which are related to the discourse on democracy or may be viewed as crucial component of it.[30] Selected categories are given by their code number, name, a short description and an example sentence in Table 3.15.

[28] https://manifestoproject.wzb.eu

[29] For methodological reflections and exemplary studies of the use of MP data in political science, see Volkens et al. (2013).

[30] The dispute on defining democracy is as old as the term itself. It is not my intention to give solid definition of democracy with my selection. It rather should be seen as a selection of democracy related categories which occur reasonably often in the data to be part of a quantitative evaluation.

Table 3.15.: Democracy related categories selected from MP data together with their number of coded sentences (n). Descriptions are cited from Werner et al. (2011), the handbook of coding instructions of the project.

Code	Name	Description	Example	n
201	Freedom and Human Rights	"Favourable mentions of importance of personal freedom and civil rights in the manifesto and other countries."	"Auch die Menschen von heute haben ein Recht auf ein gutes Leben." (FDP, 1998)	2134
202	Democracy	"Favourable mentions of democracy as the 'only game in town'."	"Mitentscheiden, mitgestalten und mitverantworten: Darauf ist Demokratie angewiesen." (SPD, 2002)	1760
301	Federalism	"Support for federalism or decentralisation of political and/or economic power."	"In der Demografiestrategie spielen die ländlichen Regionen eine große Rolle." (CDU, 2013)	632
503	Equality: Positive	"Concept of social justice and the need for fair treatment of all people."	"Deshalb wollen wir eine durchlässige Gesellschaft, in der die sozialen Blockaden aufgesprengt sind und niemand ausgeschlossen wird." (Grüne, 2009)	3577
All	Democracy Meta	Categories 201, 202, 301 and 503 are put together in one meta-category, with the aim to describe a broader picture of democratic attitude.		8103

Table 3.16.: Numbers of manually coded sentences from party manifestos of eight German parties in four elections.

Year	CDU	FDP	Grüne	LINKE	SPD	AFD	PIRAT
1998	585	1718	2292	1002	1128	0	0
2002	1379	2107	1765	880	1765	0	0
2009	2030	2319	3747	1701	2278	0	0
2013	2574	2579	5427	2472	2898	73	1794

As one can easily see, selected categories capture rather broad themes which can be expressed towards various units of discussion. Hence, realization of categories within concrete sentences of the data may be encountered in varying expressions—hard conditions for machine classifiers (and human coders as well). The size of categories is varying, as well as their coherence. A short close reading on single category sentences reveals that category 201 referencing to human rights appears much more coherent in its expressions than category 503, which contains statements towards social justice in manifold topics. Category 301 encoding federalism is rather small and often contains just references to administrative entities (federal states, municipalities or the EU), but not necessarily expresses demands for federalism explicitly. I also introduce an artificial fifth category *All*, in which I aggregate units of all other four categories. This category may be interpreted as a meta category covering a broader attitude towards democracy than just single aspects of it. For CA research this combination of codes into meta-categories is an interesting option to operationalize and measure more abstract concepts.[31]

3.3.3. Individual Classification

The selected categories represent a range of different properties of the category scheme concerning size, coherence and language variability.

[31] In quantitative studies based on MP data, index construction from aggregation of isolated category counts is a quite common approach to measure more abstract categories, e.g. right-left scales of the political spectrum (Lowe et al., 2011).

Experiments conducted in the following will examine, whether machine classification is applicable to such categories for social science purposes and under the circumstance of scarce training data.

In a first experiment, I apply supervised classification on the introduced MP data. For this, the data set of ordered sentences $\mathcal{S}$ from all party manifestos is divided in two splits:

- every odd sentence is put into a training set $\mathcal{S}_{train}$ consisting of 22,257 sentences, and
- every even sentence is put into a test set $\mathcal{S}_{test}$ consisting of 22,256 sentences.

For the experiment on individual classification, I report on decisions for algorithm selection, representation of documents as features for classification, feature selection and feature weighting, before I present base line results.

Classification algorithms: For supervised classification of textual data a variety of generative and discriminative models exists—each with its individual set of parameters to tune performance with respect to the data. For text classification Naive Bayes (NB), Decision Trees, K-nearest neighbor (kNN), Neural Networks, Maximum Entropy (MAXENT) (also known as multinomial logistic regression) or Support Vector Machines (SVM) have been widely adopted approaches (Baharudin et al., 2010).[32] Although NB performs well on many document classification tasks (e.g. spam detection), I opted it out for model selection here, because its "conditional independence assumption is violated by real-world data and perform very poorly when features are highly correlated" (ibid., p. 16). Also, Ng and Jordan (2002) have demonstrated that discriminative models for classification can be expected to outperform generative models such as NB, if training data size is large enough. Although training data in QDA scenarios usually is not abundant, we can expect enough data to learn from that discriminative classifiers appear to be the right

[32] Baharudin et al. (2010) provide a comprehensive overview on different learning algorithms and feature selection strategies for text classification.

choice. Consequently, I compare two discriminative approaches which do not assume conditional independence of features and have been reported as performing (near) state-of-the-art in many text classification applications. Approaches to compare are Maximum Entropy (Nigam et al., 1999) and SVM (Joachims, 1998). Whereas SVM is widely utilized for text classification and the baseline algorithm to beat in many ML research scenarios, MAXENT is not that common. But because of SVM having been reported to perform less optimal in situations of small training data and unbalanced classes (Forman and Cohen, 2004), I decided for comparison with MAXENT as an algorithm assumed to be more robust in this situation. For both approaches a fast and mature implementation exists in form of C++ libraries, wrapped for usage in R.[33] Decisions for feature engineering, feature selection and parameter tuning described below are based on 5-fold cross validation evaluations on the entire training set.[34]

Document representation: Documents in my experiments with MP data are single sentences $s \in \mathcal{S}$ from party manifestos, annotated with one out of five code labels. For classification, documents are transformed into feature vectors, containing potentially relevant features representing their content and class association. For this, sentences were tokenized[35] and letters transformed to lowercase beforehand. Then, the following features sets were extracted from each sentence:

- stemmed unigrams (including stopwords),
- stemmed bigrams (including stopwords),
- stemmed bigrams (with stop words removed beforehand); bigrams are not added once again, if they are already contained in the feature set from the previous step.

[33] For my experimental setup in R, I used the *maxent* package (Jurka, 2012) which wraps the MAXENT library implemented by Tsuruoka (2011), and the *e1071* package (Meyer, 2014) providing a wrapper for *libsvm* (Chang and Lin, 2011).

[34] For evaluation with k-fold cross validation see Witten et al. (2011, p. 152).

[35] For tokenization, I utilized the MAXENT tokenizer of the Apache openNLP project (https://opennlp.apache.org).

Feature selection: Performance of machine classification may be improved, both in speed and prediction accuracy, by feature selection. For this, features which do not contribute to discrimination of the classes to be predicted are removed from the feature set. Studies on feature selection repeatedly report the Chi-Square statistic as well-performing method to identify significant associations between features and class labels (Baharudin et al., 2010; Yang and Pedersen, 1997). For each feature its association with the positive and the negative class is computed by the Chi-square statistic separately. If the statistic is below a threshold of 6 for both cases, I remove the feature from the feature set.[36] A further feature pruning was applied to the extracted bigram features. Boulis and Ostendorf (2005) report that often bigrams only deliver redundant information to the classifier, compared to the observation of the unigrams they consist of. These redundant information may harm classifier performance. To overcome this issue, they propose 'Redundancy-Compensated KL' (RCKL) as a selection measure. For this, the Kullback-Leibler divergence (KL, see Eq. 3.18) between class association and unigrams/bigrams is determined. For selection, RCKL of a bigram is compared to the sum of RCKL measures of its unigram components. The goal is to remove relevant bigrams if their unigrams are also of high relevancy to distinguish between classes. Only those bigrams are kept, which add more information to the feature set, than its unigram components.

Model selection and feature weighting: For tuning SVM and MAXENT, not only feature engineering and selection is important. Finding optimal parameters for the algorithms itself is crucial, as well. For this, I performed 5-fold cross validation to infer best parameter settings (global C and C-weights per class for SVM; L1/L2 regularizers for MAXENT). Especially the C and C-weights settings were decisive

[36] Assuming degree of freedom $v = 1$ (because we distinguish two classes) a chi-square value of 6 corresponds to a significance level of $0.01 < p < 0.025$ for the association between class label and feature observation. The threshold 6 has been determined by cross validation as a valid choice to reduce insignificant features.

for SVM tuning.[37] Setting C-weights for the positive and negative class improved results of SVM classification drastically, as this parameter is supposed to mitigate the problem of SVMs with unbalanced classes. Following a heuristic, I set the weights inversely proportional to the frequency of the positive and negative class. In a last step *feature weighting* is applied to the pruned feature vectors. For this, feature counts were transformed into TF-IDF values.

Base line result: With these settings and optimization, classification for gaining base line results was conducted by training on the complete training set $\mathcal{S}_{train}$ and evaluating on the held out test set $\mathcal{S}_{test}$. For each of the selected codes of the MP data, sentences are binary classified as belonging into the category or not. Results for both classifiers are given in Table 3.17 by the conventional measures *precision*, *recall*, F_1 and *accuracy* (Witten et al., 2011, p. 163ff). We can see at the first glance that *accuracy* appears to be surprisingly high. This is an effect of the highly unbalanced classes, leaving it as a not very meaningful evaluation measure for our purpose. Comparing both algorithms we can see that SVM outperforms MAXENT in all five cases, considering the F_1 measure. While precision of MAXENT is comparable to SVM or in some cases even a little better, its recall is rather poor. Altogether, results are comparatively moderate ($F_1 < 0.5$ in four cases, $F_1 = 0.518$ in only one case for SVM). Contrasted to common ML applications which report values of $F_1 > 0.7$, these results would probably be considered unacceptable. This is clearly an effect of the very hard conditions of this classification task, described earlier (see Section 3.3.1). But firstly, we might improve the results by introducing more semantic features, and secondly, due to our changed goal of classification for trend analysis instead of individual classification, these results still might be useful.

[37] The regularization parameter C for SVMs can be seen as a penalty factor for classification errors during training. Larger C values lead to improved separation of data points of both classes by the hyperplane. But, this may also lead to overfitting to the training data. For the very sparse feature vectors of sentence classification with QDA codes, cross validation usually suggests rather small values of C as best choice.

Table 3.17.: Base line classification results: Using unigram and bigram-features, feature selection (chi-square) and feature weighting (tf-idf) yield rather mediocre classification results under the tough conditions of this task (very sparse feature vectors from single sentences $\mathcal{S}$, small amount of positive category representatives $\mathcal{S}_+$).

		SVM				MAXENT			
Code	$\mathcal{S}_+$	**P**	**R**	$\mathbf{F_1}$	**A**	**P**	**R**	$\mathbf{F_1}$	**A**
201	1083	0.359	0.556	0.436	0.930	0.336	0.373	0.354	0.933
202	903	0.350	0.592	0.440	0.938	0.430	0.348	0.385	0.954
301	326	0.457	0.331	0.384	0.984	0.242	0.196	0.216	0.979
503	1780	0.339	0.512	0.408	0.881	0.310	0.340	0.324	0.886
All	4092	0.486	0.554	0.518	0.810	0.551	0.364	0.438	0.828

3.3.4. Training Set Size and Semantic Smoothing

To approach a proper classification process for trend analysis of QDA categories, I further investigate the influence of training set sizes on the process. In a second step, the lexical feature set of uni- and bi-grams is enhanced by features generated from an unsupervised topic model process on the entire data set pushing the approach into direction of semi-supervised learning (Xiaojin Zhu, 2008). These features provide a 'semantic smoothing' on the very sparse feature vectors, improving classification quality especially for situations of small training sets.

Training set size: Baseline results in Table 3.17 suggest that training set size is significantly influencing overall performance of classification. Best results are achieved in case where all four codes are combined into the meta category 'All'. In this case, the number of positive examples $\mathcal{S}_+$ in training data is higher compared to classification of single codes, as well as the ratio of positive to negative examples is less unbalanced. However, for application of classification in QDA scenarios, generating positive training examples is very costly. Usually, the process of manual text coding involves reading through each sentence of the selected example texts for highlighting or extracting snippets fitting to categories of interest. Dependent on

the prevalence of the category, coders have to read through several hundreds of documents to obtain enough positive training examples. For negative training examples, this is considerably easier, as we might assume that every sentence not labeled with a code while looking for positive sentences is irrelevant with respect to the code book, hence a negative example.

To further compare both classification algorithms under investigation and to enlighten the influence of training set sizes, I ran classification experiments on varying training set sizes. For this, I drew random samples of size $n \in \{25, 50, 100, 200, 500, 1000, 2000, 5000\}$ from the training set,[38] and evaluated classification performance on the test set. This process was repeated 10 times. Figure 3.11 plots mean values of the 10 F_1-measures for code 'All' of both, SVM and MAXENT classification. We can observe that for smaller training set sizes $n <= 1000$ MAXENT performs slightly better than SVM. At the same time, overall performance reaches values around $F_1 = 0.4$ only, if training set sizes are above 2000 examples. This appears to be a large training set at first glance, but we know already that training set sizes are highly unbalanced. As the share of positive examples on all sentences in the training set for code 'All' is about 18 %, we can expect around $2000 \times 0.18 = 360$ positive sentences in training sets of size 2000. In a real-world QDA scenario, manual coding of 360 positive example sentences for a category is absolutely manageable. Moreover, as stated above, negative examples practically come at no cost during this process. The F_1-result for code 'All' on the entire training set of size 22,256 (see Table 3.17) also makes clear that even with a lot more training examples the F_1-measure only increases up to 0.51, suggesting that generating much more training data might be inefficient.

Semantic smoothing: In Section 3.3.1, I have stated that feature vectors originating from the 'bag-of-words' model on single sentences or paragraphs are extraordinary sparse, contributing to low perform-

[38] The drawing of a random sample was repeated if there was no single positive example in the draw which may happen often for the very small training set sizes.

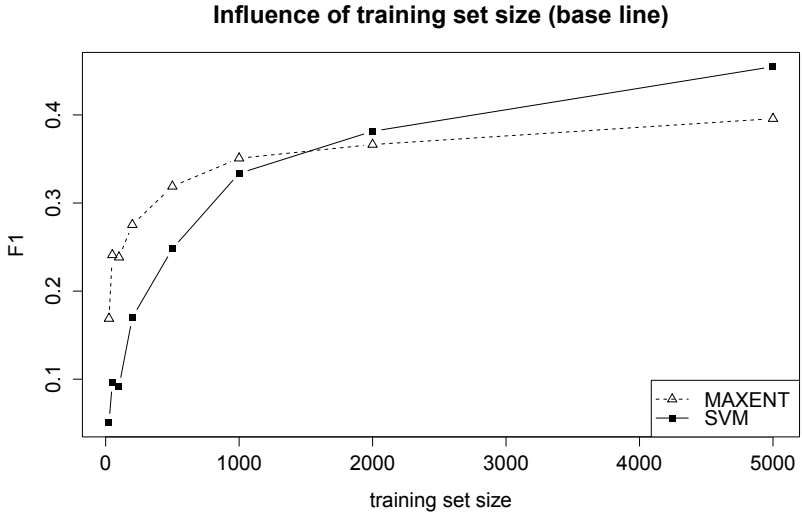


Figure 3.11.: Influence of training set size on classification performance of two classifiers, MAXENT and SVM. Classification is based on word features (uni-/ bigrams) for the meta-category *'All'*.

ance in classification processes. Based on theoretical assumptions presented in Chapter 2, I also described the importance of context and latent semantics for ML applications. To improve classification quality in sparse feature vector situations, consideration of context seems to be a valid approach. In this respect, several suggestions based on clustering semantics in unsupervised manner to extend available information for supervised learning have been made in ML literature, leading to the paradigm of semi-supervised learning (Xiaojin Zhu, 2008). For NB classification, Zhou et al. (2008) propose an approach for "semantic smoothing". They introduce a concept of 'topic signatures' to augment observations of lexical features given a class. 'Topic signatures' might be seen as a representation of latent meaning implied by word observations, which can contribute to classification by additionally taking 'topic signatures' given class into account. With

their approach Zhou et al. improve generative NB classification, especially for small training data sets. For discriminative classifiers, a different, more general approach for 'semantic smoothing' needs to be obtained. Phan et al. (2011) introduce a generalized approach to generate features for supervised classification of a document collection $\mathcal{D}$ by employing LDA topic modeling on an unlabeled (universal) data set $\mathcal{W}$ (e.g. Wikipedia corpora).[39] The idea can be summarized in the following steps:

1. collect a universal data set $\mathcal{W}$ (e.g. Wikipedia articles or, if large enough, your collection under investigation),
2. compute a topic model with K topics on $\mathcal{W}$,
3. for each document $d \in \mathcal{D}$ use the topic model from step 2 to sample topic assignments θ_d for the words it contains, without updating the $\beta_{1:K}$ parameters for term distributions per topic,[40]
4. convert counts of topic assignments in d into K additional features for classification.

As an LDA topic model may be seen as an overlapping clustering of general senses or meanings, assigning new documents to these clusters enriches documents with some latent semantic information, which contributes as smoothing of its very sparse word features. I applied the framework of Phan et al. (2011) for improving the classification performance on the MP data set. As universal data set $\mathcal{W}$, I utilized the party manifestos of the MP data itself. For this, I put sequences of every 25 sentences from $\mathcal{S}$ into one pseudo-document (1st step) which serves as input collection to compute the topic model (2nd step). On this collection, a model with $K = 50$ topics was computed by

[39] An early, but less systematic realization of this idea can be found in Banerjee (2008).

[40] This proceeding is also applied for online topic modeling of document streams (Yao et al., 2009). There also, the model is initially calculated on a fixed population. Then, topics for documents from the stream are sampled on the initially computed β distribution without updating it.

1,000 iterations of Gibbs Sampling ($\alpha = 0.02, \eta = 0.002$). A short qualitative investigation of the most probable terms of each topic shows expected results: inferred topics represent semantic clusters related to various policy issues important in the last German elections. In the third step, the collection to be classified is again the set $\mathcal{S}$ of sentences from the MP data. For every sentence $s \in \mathcal{S}$, topics are assigned to all words it contains by 100 iterations of Gibbs Sampling of an LDA process initialized by the $\boldsymbol{\beta}$ parameter of the model computed in the previous step. Since sentences can be very short, results of topic assignments to words in s can be very unstable. To get more reliable results, I repeat topic inference on s 20 times. Counts of topic-word assignments in s are averaged over these 20 runs, before they are converted into 50 additional features for each sentence.

Figure 3.12 shows the average results of 10 iterations of the classification with these additional features for increasing training set sizes. For both classifiers, we observe that additional features from the topic model improve the performance up to 5 percentage points. Furthermore, improvements get lower, as training set sizes increase. If all training data available is taken into account, there is almost no performance gain of this method compared to the baseline (see Tables 3.18 and 3.17). This is due to the fact that the effect of semantic smoothing diminishes, the more training data is available to the classifier. In one case, code 301 (*federalism*) performance even considerably decreases. This is probably a consequence of the heterogeneous nature of this rather small category. Statements in favor of federalism usually come with a variety of different policy issues. Narrowing the category to certain semantic topics by topic model features improved the precision, but lowered the recall on the test set. This is an important hint towards the need for precise and coherent category definition and application during manual coding. Nonetheless, in our scenario for QDA application of ML the method of semantic smoothing provides essential improvements for the other four categories. We can expect improvements for QDA application in general, because we usually operate on small training data.

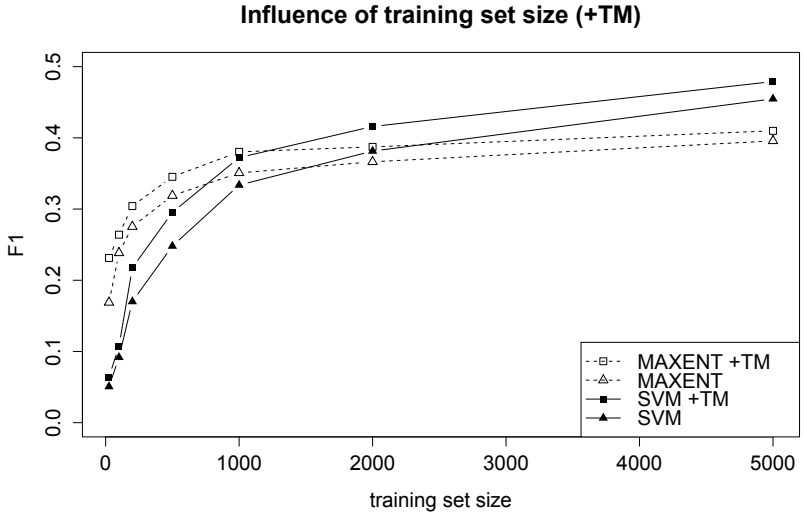


Figure 3.12.: Influence of training set size on classification performance of two classifiers, MAXENT and SVM for the meta category *'All'*. Classification is based on word features (uni-/bigrams), and additionally on features generated by topic model inference on each sentence (+TM).

Table 3.18.: Semantic smoothing classification results: features from topic modeling additionally to uni-/bigrams improve classification performance most for small training sets. If all training data available is used (this Table), performance gain compared to the base line (see Table 3.17) diminishes.

Code	$\mathcal{S}_+$	**P**	**R**	**F**	**A**
201	1083	0.3567	0.5152	0.4216	0.9312
202	903	0.3791	0.5437	0.4467	0.9453
301	326	0.5145	0.1625	0.2470	0.9854
503	1780	0.3440	0.4865	0.4030	0.8847
All	4092	0.5011	0.5733	0.5348	0.8166

3.3.5. Classification for Proportion and Trend Analysis

Previous sections have shown rather mediocre results of classification performance for the content analysis scenario on the MP data set. For smaller training set sizes using an optimized SVM with unigram, bigram and topic model features, we can expect F_1-values around 0.4 to 0.5 at best. If individual classification had been the goal of our classification scenario, these results would have been rather unacceptable. Dissatisfaction with the computationally evaluated results can be somewhat mitigated by having a close look on the false positives during classification. Often these are not really false positives in the sense of not fitting the description of the category in the code book. In fact, they are often ambiguous statements which are just labeled with another label, stressing a different aspect of the manifesto sentence. Although in these cases it would make sense, multiple labels are only annotated in rare cases in the MP data set. In conclusion, for the goal of individual classification further effort would be useful to improve the category system, the annotated data set and the classification workflow (including feature engineering and feature selection).

But instead of valid individual classification, I defined valid prediction of proportions and trends in diachronic data as primary goal of the classification process in 3.3.1. For this, we investigate if the moderate individual classification performance achieved so far still might be sufficient to produce valid and reliable results towards these goals. Accordingly, we need to change our evaluation criteria. In addition to precision, recall and F_1, we assess classification performance by:

- **Root Mean-Square Deviation (RMSD)**: individual class labels assigned to documents in the test set can be counted as class proportions—the share of a class on the entire test set. Splitting the entire set into single manifestos (one document per party and election year) yields multiple measurements for each class proportion in these subsets. $RMSD = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(x_{1,t} - x_{2,t})^2}$ is an established measurement to assess the deviation of predicted class proportions

$x_{1,\cdot}$ to actually observed class proportions $x_{2,\cdot}$ in a time series with n data points, i.e. proportions in single manifestos. As we compare proportion values ranging between zero and one, we may interpret RMSDs (which consequently also have a range [0, 1]) as error on the estimation missing the true proportion value of a category.

- **Pearson product-moment correlation (r)**: classification quality for trend analysis can be determined by measuring the association between predicted and actual quantities of class labels in time series. Again, we assume splits of our predicted and actual test set labels into single manifestos as two time series $x_{1,\cdot}$ and $x_{2,\cdot}$. If increase and decrease in absolute frequency of positive class labels or relative class proportion go along with each other, we expect a high Pearson product-moment correlation (Pearson's r). Significance of Pearson's r can be assessed by a statistical test.

The selected MP data contains manifestos from eight parties and four elections (see Table 3.15). For the following experiments, I treat the parties PDS and DIE LINKE as the same, as the latter has been founded as a merger of the PDS and the WASG, a second left-wing party in Germany in 2007. AFD and PIRATEN did not come up before elections in 2013. All in all, this gives us a split of the overall test set into 22 data points to determine the absolute number or the relative share of code labels in the actual data and compare them to the classifier's prediction. On these 22 test set splits, two different approaches of category proportion estimation are applied and evaluated by RMSD and Pearsons's r.

Estimating Proportions from Feature Profiles

Hopkins and King (2010) propose a method of "proportional classification" for CA, optimized for social science goals. Their method does not rely on aggregated individual classification predictions to measure category proportions in a population of documents. Instead of counting predicted labels for individual documents, they estimate proportions of categories in a test set by observing probabilities of

"word stem profiles" S in the entire training set and test set. Such word stem profiles are defined as binary vectors encoding the presence or absence of unigram features in a document. For a vocabulary of size $K \leftarrow |V|$ word stems, there exist 2^K different profiles (i.e. the power set of the feature set). For each document its corresponding feature profile can be determined. In practice, because feature space is large, profiles on all features would be too numerous and mostly unique. Therefore, the procedure relies on subsets of features, e.g. $n = 1000$ repetitions of random draws of $K = 10$ features out of the entire feature set V. For a document collection $\mathcal{D}$ a multinomial distribution $P(S)$ with 2^K values, encoding probabilities of occurrence for each feature profile in the collection can be determined. Marginal probabilities of profiles $P(S)$ and probabilities of profiles given classes $P(S|\mathcal{C})$ can be observed directly in the training data. Because $P(S) = P(S|\mathcal{C})P(\mathcal{C})$ where we know already two of three terms in this equation, probabilities (i.e. proportions) of labels $P(\mathcal{C})$ in a test set can be determined by standard regression algebra[41] under the assumption that conditional probabilities of feature profiles given classes $P(S|\mathcal{C})$ in the training set and in the test set are the same.

This method provides very accurate estimates of category proportions in unknown document populations, as evaluations by Hopkins and King (2010) show. Hence, it seems reasonable to employ their approach for proportion and trend detection on the MP data as well. To evaluate the performance on the MP data set, I conduct two experiments:

1. proportion estimation in the entire test set $\mathcal{S}_{test}$,

2. proportion estimation in test set splits of single manifestos for valid trend detection.[42]

[41] To solve the regression model $\mathbf{y} = \mathbf{X}\lambda$ (without any error term) for λ we need to calculate $\lambda = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ (Hopkins and King, 2010, p. 236f).

[42] Hopkins and King (2010) provide the R package "readMe" as reference implementation to their paper. But because the method is rather simple and I needed slight modifications for applying it to the second experiment, I re-implemented it on my own.

Table 3.19.: Hopkins/King method of proportion and trend prediction evaluated by RMSD and Pearson' r: While error rates on the entire test set are very low, predictions for subsets of it are fairly inaccurate. Hence, trend predictions (r), although significantly positive, are not very exact either.

Code	RMSD (test set)	RMSD (splits)	r (proportions)
201	0.0041	0.1537	0.7900
202	0.0092	0.0987	0.4908
301	0.0066	0.0225	0.6049
503	0.0117	0.0723	0.7787
All	0.0116	0.2319	0.5505

Table 3.19 displays the results for these two experiments on the MP data set. They confirm that the method proposed by Hopkins and King provides valid estimations of code proportions in our test set. Estimations for the entire test set are very accurate, only producing an error around 1 percentage point for all categories classified. However, the test set represents a sample selected from the entire MP data set by the same strategy as the training set (every odd/even sentence). Hence, distributions of features may be assumed as almost identical in each of the disjoint sets. This does not hold true if we split the test set deliberately into separate manifestos of the single parties per election year. For this, averaged RMSD values are given in the third column of Table 3.19. Here, results indicate immense discrepancies between estimated and predicted proportions in four out of five categories. The method heavily over- or underestimates proportions for the codes 201, 201, 503 and 'All'. A closer look into the data reveals that overestimations seem to correlate with high relative shares of the category in certain party programs. For example, the party PDS/LINKE has a high share of sentences expressing demands for 'social equality' (code 503). As the regression model instance calculated $P(S|\mathcal{C})$ on the basis of the entire training set (instead of sentence only from PDS/LINKE), the higher relative share of feature profiles $P(S)$ associated with 'social equality' in PDS/LINKE manifestos yields to overestimation of

the category in their manifestos. In contrast, shares of this category in manifestos of the conservative party CDU are underestimated according to the lower share of feature profiles associated with this category compared to the entire training set of all parties. From this, we may assume that parties use their own specific vocabulary to express ideas on the same aspect of democracy in their very own words.

The crucial assumption that the association between feature profiles and class proportions $P(S|\mathcal{C})$ in the training data may as well be assumed for the test set, does not apply if the test set is a subset with a biased distribution of feature profiles. This makes the Hopkins/King model very vulnerable to altered distributions of feature profiles in sub-sets of the entire collection. To circumvent this effect, we would need to compute a new model instance $P(S|\mathcal{C})$ from subsets of the training data consistent with the splits of the test data, i.e. also split our training data by party and year. But then, training data size for each split will considerably decrease while costs for model training for time series estimation will increase drastically. All in all, the Hopkins/King model estimates proportions very accurately in situations where "among all documents in a given category, the prevalence of particular word profiles in the labeled set [... is] the same in expectation as in the population set" (ibid. p. 237). Yet, as soon as word profiles in the training set are not an (almost) identically distributed subset from word profiles of the target population, the model is unable to provide accurate estimations any longer.[43] Thus, for time series analysis, where we want to estimate category proportions in different time sliced subsets, the method becomes impractical. Consequently, correlations between predicted and actual category shares in the test set splits (column four of Table 3.19), although

[43] To further confirm this, I also conducted an experiment where I do not employ meaningful test set splits by party manifestos, but by random draws from the test set. Random draws guarantee independent and identically distributions of feature profiles in the test set splits. Results were as expected: If test set splits are random subsets from the entire test set, estimations of category proportions were rather accurate again.

Table 3.20.: Individual classification aggregation method of proportion and trend prediction evaluated by RMSD and Pearson's r: Measures evince that prediction of proportions and trends can be quite accurate ($r > 0.9$) although the corresponding F_1-measure is rather moderate.

Code	F	RMSD	r (counts)	r (proportions)
201	0.4216	0.0258	0.9593	0.9221
202	0.4467	0.0246	0.9639	0.9106
301	0.2470	0.0133	0.7504	0.5785
503	0.4030	0.0413	0.9685	0.8157
All	0.5348	0.0483	0.9836	0.9009

indeed positive and statistically significant, are not overly high that we might assume a correct time series predictions. It remains to be seen whether aggregated individual classification is able to provide us with more reliable estimations in this respect, if it is provided with good training examples.

Aggregating Individual Classification

The optimized SVM classifier with its topic model enhanced feature set (Section 3.3.4) already classified each sentence in the test set either belonging to a category or not (see Table 3.18). Having a predicted label for each sentence in the test set, error rates on proportion estimation and trend predictions on single manifestos are directly observable. Table 3.20 gives evaluation measures for supervised classification of proportions and trends for all five codes under investigation compared to the classic F_1-measure.

For the different codes classified, RMSD is not lower than 1 percentage point and not greater than 5 percentage points. Compared to the previous estimations by the model of Hopkins and King (2010), error rates for proportion estimations on the entire test set are noticeably

higher. Still, error rates are not unacceptably high.[44] Nevertheless, as the share of sentences of a certain code in the test set is very unbalanced, deviations of some percentage points indicate significant over- or underestimation in absolute numbers. For example, category 201 has only a share of 4.86 percent in the entire test set. An RMSD of 0.025 indicates an average over- or underestimation around 2.5 percentage points, or 50 percent in absolute numbers of sentences classified as belonging to category 201. We can conclude that on the one hand, estimations of proportions are rather stable and do not deviate heavily from the global perspective on the entire data set investigated. On the other hand, we need to be careful by assessing on exact numbers of proportion quantities as small numbers of deviations on proportions may entail significant over- or underestimation in absolute numbers of classified analysis units.

Evaluation on trend correlation is more promising. Instead of exact estimation of proportions, we judge the quality of the classification process only by its ability to predict increases or decreases of quantities in time series correctly. Correlation coefficients for absolute counts are very high: $r > 0.95$ for four out of five codes. If correlation is based on estimated proportions instead of absolute counts we still obtain very high correlations ($r > 0.9$ in three cases). Judging trend prediction on relative proportions is preferable, because correlation between absolute counts and predictions is not only determined by the quality of the classifier, but by the size of the test set splits as well. If there are more sentences in a split, it may be assumed that chances are higher for more sentences to be classified positively.

As noticed earlier, category 301 (federalism) appears to be problematic due to the very small number of training examples and heterogeneity of its content. Except for this category, trend correlation of individual classification significantly outperforms correlation based on proportional classification with the Hopkins/King approach by large extent. In contrast to the latter, the SVM model is able to generalize

[44] Hopkins and King (2010) state that in comparison to survey analysis with random sampling on a national level, RMSDs up to four percentage points are not considered as unusual (p. 241).

its information learned from the training set, to predict proportions in arbitrary test sets reliably well—a finding that also has been reported by Hillard et al. (2008). The fact that for trends on relative proportions we may obtain correlations of $r > 0.9$, although the F_1-measure with values around 0.4 is moderate only, is an important extension of that finding. It shows that even with small F_1-values, we are able to predict trends in diachronic data correctly.

The connection between conventional evaluation measures for text classification (F-measure, precision, recall) and the two newly introduced evaluation criteria can be investigated further experimentally. Figure 3.13 plots the correlation coefficient r and RMSD in dependency of different levels of precision and recall from classification of the code 'All'. Varying precision/recall ratios are introduced artificially by varying the threshold of probability values, in which case to assume a positive code label. The SVM classifier can provide a probability value for each predicted instance $s \in \mathcal{S}_{test}$.[45] Usually, if the probability $P(+|s)$ of a positive label given s is higher than threshold $t = 0.5$, the classifier attaches the label to the instance. By changing this probability threshold t within a $(0, 1)$ interval, lower values for t increase recall while decreasing precision. Higher t values have the opposite effect. The interesting observation from this experimental setup is that the correlation coefficient r as well as RMSD reach very good result levels over a wide range of t. From this, we may infer that classification errors of the supervised SVM approach do not lead to arbitrary false predictions of trends. Even if the ratios between precision and recall change drastically, estimation of trend correlations remain stable. At the same time, we observe optimal r and RMSD measures, in case

[45] Actually, the SVM classifier infers a separating hyperplane in the feature space based on the training data which allows for deciding whether an unlabeled data instance lies inside or outside of the region of the positive class. This is indicated by the classifiers output of the margin to the hyperplane, i.e. 0 indicates values located exactly on the hyperplane, values above 0 indicating the positive class and values below 0 the negative class. Margin values can be transformed to probability values by applying the method of Platt scaling (Platt, 2000). Scaled data instances located near the hyperplane with margins around 0 correspond to probability values around 0.5 for positive labels.

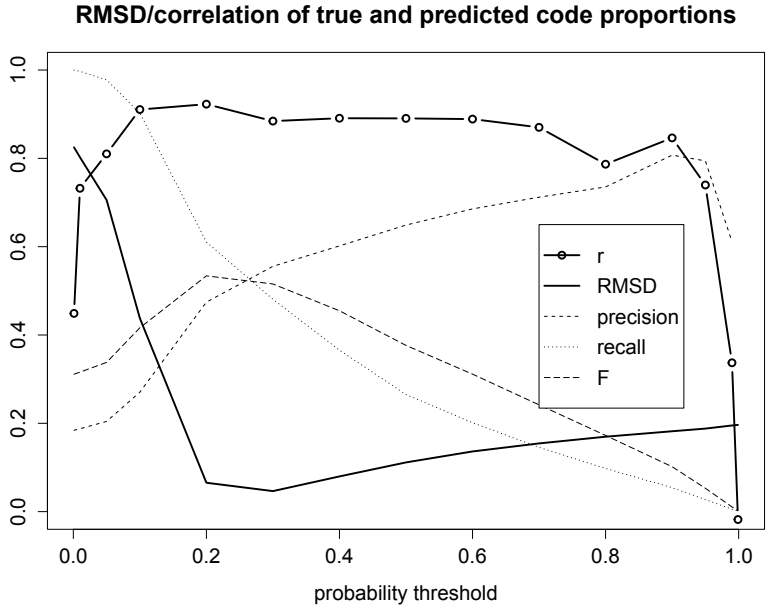


Figure 3.13.: RMSD and correlation dependent on different levels of precision and recall.

the F_1-measure is highest. Thus, optimizing a classification process with respect to the F_1-measure clearly is a worthwhile strategy to obtain valid results for trend and proportion analysis. Nonetheless, we do not need to push it into regions of $F_1 = 0.7$ or higher to get acceptable results for our QDA purposes.

To visualize the classification performance for trend detection, Table 3.21 displays classifier predictions and true values for absolute counts and relative proportions of the investigated codes for the five major German parties during the last four elections. These plots confirm visually the numeric evaluations of high correlations between automatic retrieved and actual (manually labeled) category quantities. The classifier tends to label more instances as positive for a code

than there are actually in the test set.[46] In a last step, we want to investigate how to collect good training data for automatic CA efficiently.

3.3.6. Active Learning

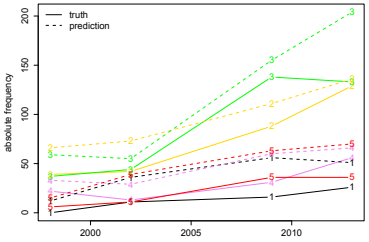
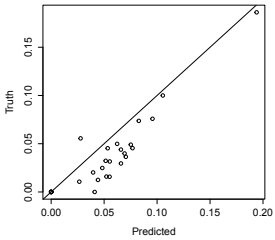
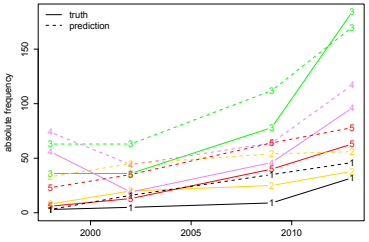
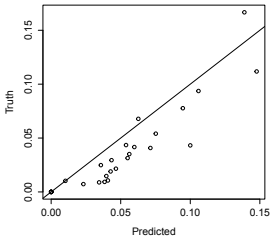
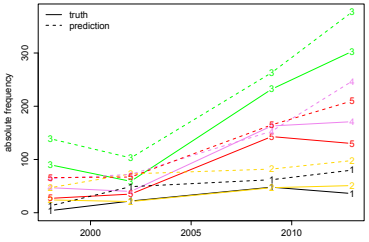
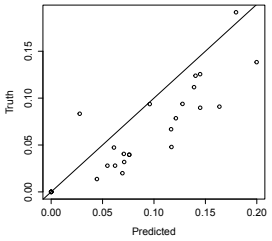
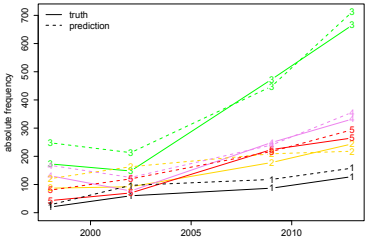
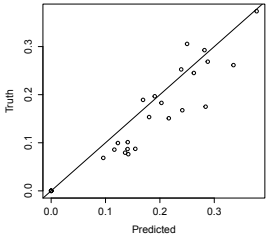
Experiments to evaluate classification performance have been conducted on the entire training set so far. This includes between 326 (code 301) and 4092 (code 'All') positive training sentences per code (see Table 3.17). Collecting several hundreds or even thousands of positive training examples for CA is very costly. Analysts need to read through many documents and code positive analysis units for each category manually. To support this work, I introduce in the final step of our classification experiments a workflow for the efficient production of training data with high quality for trend analysis.

For efficient training data collection we can employ the active learning paradigm: "The key hypothesis is that if the learning algorithm is allowed to choose the data from which it learns [...] it will perform better with less training" (Settles, 2010). The basic idea is to start with a little training set $\mathcal{S}$ based on manual reading and coding. This initial training set is then augmented by new instances, which are suggested by supervised classification of the set of unlabeled analysis units $\mathcal{U}$. Suggestions, so called *queries* of the active learning algorithm, have to be evaluated by an *oracle*, e.g. a human annotator, who accepts or rejects them as a positive example for the category of interest. After evaluation of suggested queries, supervised classification is performed once again on the training set extended by the newly reviewed examples.

Active learning scenarios might be distinguished into stream-based and pool-based (ibid.). In the stream-based scenario, the algorithm

[46] A short manual investigation of the false positives reveals that retrieved sentences are often not really bad examples for the category of interest, if judged by the code book. Again, this is a hint to carefully craft categories and apply code books during codification. Beyond classifying unlabeled data, this process also might be utilized to improve the quality of the already labeled training data by revising alleged 'false positives' on held out training data.

Table 3.21.: Left column: Estimates and true values for counts of coded sentences in manifestos of the five major German parties (1 = CDU, 2 = FDP, 3 = Grüne, 4 = PDS/LINKE, 5 = SPD). Right column: Estimated against true proportions.

Code	Absolute Counts	Proportions
201		
202		
503		
All		

Workflow 1: Pool-based batch-mode active learning

Input:
Initial set $\mathcal{S}$ of manually labeled sentences
Set $\mathcal{U}$ of unlabeled sentences
Query selection strategy q
Batch size n
number of maximum iterations $iMax$
Output:
$n \times i$ new high quality training examples

```
1 i ← 0
2 while i < iMax do
3 |   i ← i + 1
4 |   Train model on S
5 |   Apply model on U
6 |   Rank sentences in U by strategy q
7 |   Manually label top n sentences
8 |   Move labeled sentences from U to S
```

decides for every unlabeled data instance in $\mathcal{U}$ individually whether it should be presented as a query to the oracle. For this, it employs some kind of 'informativeness measure' on the data instance to reveal, if it lies in a region of uncertainty of the feature space. In the pool-based scenario, the algorithm first ranks all unlabeled data instances in $\mathcal{U}$ according to their informativeness, and then selects the best matching data instances for querying the oracle. Pool-based query selection appears to be much more common among application scenarios (Settles, 2010, p. 12). It can further be distinguished into serial and batch-mode active learning (ibid. p. 35). In the former only one query is evaluated per iteration, while in the latter a set of n best matching queries is selected for evaluation, before a new iteration is started (see Workflow 1). This strategy is advisable, if costs for training the classifier are high or multiple annotators should evaluate on queries in parallel. Hence, the pool-based batch-mode scenario of

active learning is perfect for our application to develop an efficient training data generation workflow for QDA classification.

As 'informativeness measure' to select queries from unlabeled sentences $u \in \mathcal{U}$, I simply decide for the positive category probability the SVM can provide when predicting a label for u based on the current training set $\mathcal{S}$. As probability suggests, the region of uncertainty lies around values of $P(+|u) = 0.5$. The active learning process for our task is then influenced by three parameters mainly:

1. *Query selection strategy:* how should queries be selected from the pool of unlabeled data, to a) minimize evaluation efforts of the oracle, and b) maximize classifier performance with respect to valid trend prediction?

2. *Size of the initial training set:* how many training examples should be collected before starting active learning to guarantee the goal of valid trend prediction in time series data?

3. *Probability threshold:* a threshold on the classifier's output of the probability for assigning a positive label to a data instance may influence the pool-size where queries can be selected from. Above which probability threshold a data instance should be considered as query candidate during an active learning iteration?

In the following experiments, I simulate the active learning procedure to investigate the influence of query selection strategies as well as initial training set sizes and probability thresholds for the process. For each category to classify, I initiate the learning process with a random selection of $a = 100$ positive training sentences and the same amount of random negative examples. In every following iteration the $n = 200$ best sentences, according to a certain selection strategy, together with their true labels are added to the training set. Adding the true labels mimics the oracle decision of query evaluation usually done by human annotators. During every iteration step, acceptance rate (number of evaluated queries as positive), F_1-measure of 5-fold cross-validation on the training set, F_1-measure on the test set and Pearson's correlation

r on the test set splits of single manifestos are calculated, to judge on the improvement while learning. Evaluation measures are plotted in Figure 3.14. Additionally, F_1 and r from previous experiments on $\mathcal{S}_{test}$ utilizing the entire training set $\mathcal{S}_{train}$ (see Table 3.18) are given as reference values—both are drawn into the plots as horizontal lines. They allow to visualize how evaluation criteria approach results of optimal training data situations very early during the active learning process with small training data sizes already.

Query selection strategy: Three query selection strategies are tested and compared. During each iteration of active learning, sentences from the so far unlabeled data set $\mathcal{U}$ are selected by

1. *Certainty:* highest probability of belonging into the positive category
2. *Uncertainty:* proximity to the decision boundary $t = 0.5$ of the probability belonging into the positive category,
3. *Random:* sampling from all sentences above a probability threshold $t = 0.3$ in $\mathcal{U}$.

Figure 3.14 displays the progress of learning with the three different query selection strategies on the code 'All' during 10 iterations. Main evaluation criterion for the strategies is, how the selected training examples perform on predicting trends in the test set correctly (dotted black line). Visually we can determine that relying on the most certain examples for active learning does not improve the classification towards the goal of trend prediction very well. Although the acceptance rate of queries (solid circled line) is highest compared to the two other strategies, examples selected provide rather redundant information to the classifier instead of learning new, so far ambiguous information. Relying on uncertain examples instead (strategy 2) slightly lowers the acceptance rate, but improves trend correlation. We need one more iteration, to collect 400 or more positive training examples (marked by the vertical red line). But these training examples certainly describe better the decision boundary between the positive and the negative

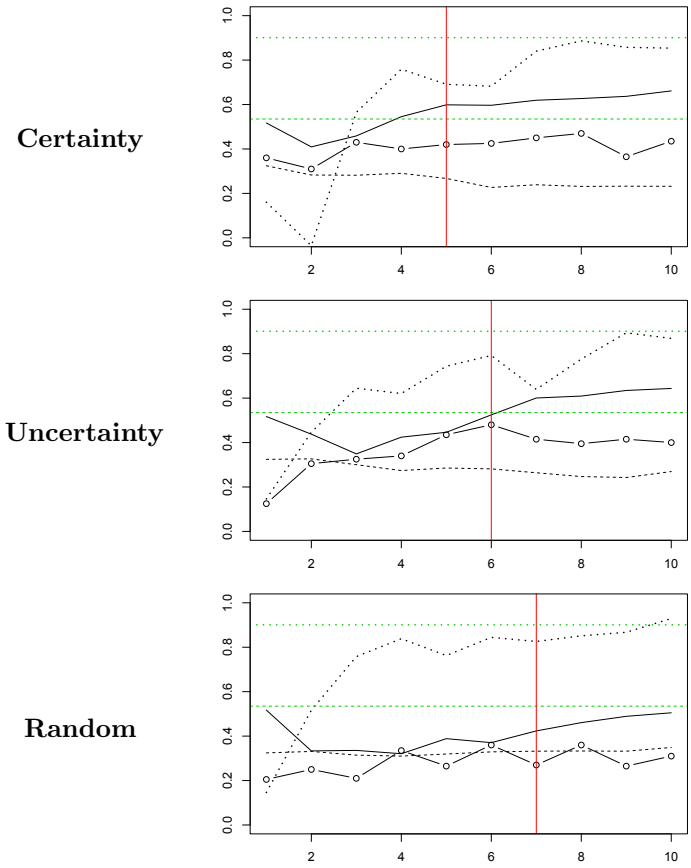


Figure 3.14.: Performance of query selection strategies by ongoing iterations of active learning (x-Axis): F_1 on the test set (dashed black line), F_1 of 5-fold cross validation on the current training set (solid black line), acceptance rate of queries (solid, circled line) and Pearson's correlation r between predicted and true label quantities on test set splits (dotted black line). Green horizontal lines indicate F_1 and r as reference, when the entire sets $\mathcal{S}_{train}$ and $\mathcal{S}_{test}$ are used. The vertical red line marks the iteration, when 400 or more positive training examples are collected.

class of the category 'All'. Nevertheless, just ranking and selecting queries by uncertainty also includes redundant examples centered around the decision boundary. Homogeneity of the iteratively learned training set is also suggested by high rates of F_1-measures (solid black line) for 5-fold cross validation on the learned training set. After four to six iterations, their value exceeds the reference value for the F_1-measure on the entire sets $\mathcal{S}_{train}$ and $\mathcal{S}_{test}$ from previous experiments (horizontal dashed green line).

Redundancy and undesired homogeneity of the learned training set are mitigated only by the third strategy of randomly selecting queries $u \in \mathcal{U}$ with a positive class probability of $P(+|u) >= 0.3$. For all strategies we can observe, when collecting training data long enough, reference values for trend correlation (i.e. using the entire training data set) are reached (dotted green line). At the same time, F_1-measures on the entire test set $\mathcal{S}_{test}$ (dashed black line) remain with values between 0.3 and 0.4 significantly below the reference value. Again, this is a strong indication for the fact that we do not need overly accurate individual classification, to perform valid trend prediction.

We also would like to know how many positive examples we need to collect, until we can expect a valid estimation on proportions and trends in the data. Unfortunately, this is hard to answer in general. There are some approaches of defining "stopping criteria" for active learning processes (Vlachos, 2008),[47] based on the idea that the process should stop, if no queries could be identified in the unlabeled pool that would add significantly more information to the classifier than it already contains. However, these approaches seem to be rather impractical for CA purposes. Because of language variety, we still can

[47] Vlachos (2008) suggests to use certainty measures of classifiers on an unlabeled held out data set to define a stopping criterion. For SVMs certainty can be defined as averaged absolute margins of classified instances to the separating hyperplane. Average margins of instances in held out data should increase during active learning iterations up to a certain point due to rising certainty based on more training examples. If there are no longer examples in the pool of unlabeled training data which provide new information to the SVM, classifier certainty on held out data is supposed to decrease.

find new informative examples after many iterations. Settles (2010) also states: "the real stopping criterion for practical applications is based on economic or other external factors, which likely come well before an intrinsic learner-decided threshold" (p. 44).[48] At the same time, our classifier might be able to predict trends correctly, based on the information learned at a much earlier point of the process. To keep the effort manageable, I will provide a rule-of-thumb as stopping criterion based on the training set size of positive training examples, as well as the number of learning iterations. Hopkins and King (2010) suggest to collect not more than 500 training examples to get accurate estimations of category proportions. Since we are interested in measuring certain code book categories realized in sentences in our data, we should instead concentrate on the number of positive examples of a category than on the whole set of annotated examples, positive and negative altogether.[49] During experimentation, I observed that collecting around 400 positive examples was sufficient in all cases, to provide reliable estimates of trends and proportions in the MP data. This is also a manageable number of examples to collect. Hence, I decided for 400 examples as a reference goal in the active learning process.

Measuring trend correlations at a point when 400 or more positive training examples have been collected allows for strategy and parameter comparison beyond visual display of the learning curves. As results of this simulation heavily depend on random initialization of training examples—in case of query selection strategy 3 also on random selection of active learning queries—the procedure is repeated 10 times for every code. Results of the 10 runs are averaged and

[48] In an experiment I conducted on the MP data set, average margins of the SVM started to decline after the 14th or 15th iteration of evaluating batches of $n = 200$ newly selected training examples. This suggests, we would need to evaluate around 3,000 example sentences for a single category until reaching the numerically advised stopping criterion. For the most applications, this appears to be too much of an effort.

[49] As mentioned earlier, during manual annotation of sentences / paragraphs, negative examples come in large numbers at low cost. At the same time, they do not contribute much to understand and define the category of interest.

Table 3.22.: Comparison of averaged 10 runs of three query selection strategies for active learning. Trend correlation as Pearson's r on the test set as well as improvements of the best strategy (random) over the other two are given. * ($p < 0.05$) and ** ($p < 0.01$) indicate statistical significance of the improvements.

Code	r (cert.)	r (uncert.)	r (rnd)	vs. cert.	vs. uncert.
201	0.8060	0.8954	**0.9108**	**13.0%	1.7%
202	0.7558	0.8927	**0.9025**	*19.4%	1.1%
503	0.6376	0.7182	**0.7422**	**16.4%	3.3%
All	0.6576	0.8058	**0.8340**	**26.8%	3.5%

evaluated by a statistical t-test to determine statistical significance of differences between the strategies. Table 3.22 displays average trend correlations and the improvement of the best strategy against the others in percent. We can observe that the random selection strategy (rnd) yields classification models which predict label quantities correlating highly in trends with the actual data already after few iterations. Although collecting positive examples quicker, the other two strategies need more iterations to collect a training set which contains sufficient good and varying examples to predict trends validly. This finding is consistent with experiments in the active learning literature on standard NLP corpora (Settles, 2010, p. 35).

Initial training set size and probability threshold: After having identified the random query selection strategy as preferred for active learning towards trend prediction, we shortly have a look on two further parameters of the process. Firstly, does the size of the initial manually labeled training set have an influence on the efficiency of the learning process? Should we start with larger or smaller quantities of training examples to provide sufficient information at the beginning of the process or to avoid selection bias of analysts? Secondly, we want to select a suitable threshold value t for the probability of a positive label as the basis for the pool of potential queries during each active learning iteration. Choosing a small threshold might

Table 3.23.: Comparison of averaged 10 runs of different initial training set sizes a and probability thresholds t for query pool selection. Initial training set sizes seem not to have a clear influence on the process. For probability thresholds there is a tendency to lower thresholds for better results. Yet, improvements are not statistically significant. $\bar{I}_{400}$ gives the average number of active learning iterations per test scenario to reach the goal of 400 positive training examples.

Code	**initial training size (a)**			**probability threshold (t)**			
	200	**100**	**50**	**0.2**	**0.3**	**0.4**	**0.5**
201	0.9070	**0.9108**	0.8717	**0.9232**	0.9108	0.8751	0.8882
202	0.8339	0.9025	**0.9042**	**0.9056**	0.9025	0.8966	0.8635
503	**0.7773**	0.7422	0.7431	**0.7556**	0.7422	0.7366	0.7287
All	0.8037	0.8340	**0.8359**	0.8212	**0.8340**	0.8088	0.7915
$\bar{I}_{400}$	6.75	7.72	7.82	9.25	7.72	6.67	5.92

produce more valid results for trend prediction, as a bigger variety of training examples has the chance to be selected from the pool. On the other hand, a too small threshold increases the number of iterations $\bar{I}_{400}$ necessary to collect the targeted goal of 400 positive training examples, since there are more queries from ranges of lower probability which actually belong into the negative class. Table 3.23 displays experimental results for variations of initial training set sizes $a \in \{50, 100, 200\}$ and probability thresholds $t \in \{0.2, 0.3, 0.4, 0.5\}$. Differences between the results of 10 averaged runs are statistically insignificant, indicating that influences of initial training set sizes and thresholds are not especially decisive for the overall process. Nonetheless, evaluation suggests that there is a tendency towards smaller probability thresholds. From this experiment we can infer that decisions on initial training set sizes and thresholds may be taken pragmatically. If there are many good examples for a category which are easy to collect, it seems to be maintainable to start with a bigger training set. If a category is expressed in fairly coherent language without much variety (codes 201 and 202), it seems absolutely valid,

to just collect a few examples to initiate the process. For probability thresholds, we can weigh between an acceptable number of batch iterations $\bar{I}_{400}$ (tendency towards higher thresholds) and a better quality (tendency towards lower thresholds). With respect to this trade-off, selecting $t = 0.3$ appears to be a reasonable default choice.

3.3.7. Summary of Lessons Learned

The section on text classification addressed a wide range of research issues from NLP in the light of their application for QDA. Conducted experiments identified reasonable solutions for this purpose. Applying supervised machine learning to the process of 'coding', i.e. assigning semantic categories to (snippets of) texts, allows for efficient inspection of very large data sets. Qualitative categories become quantifiable through observation of their distribution in large document populations. To effectively execute this, special requirements and circumstances for the application of machine classification have to be taken into consideration. For this, the previous sections suggested solutions for optimization and integration of these aspects into a text classification workflow which allows content analysts to determine category quantities in large text collections reliably and validly. Methods of classification model selection, feature engineering for semantic smoothing and active learning have been combined to create a workflow optimized for trend and proportion estimation in the data. Evaluations during single steps of the entire chain have contributed to some valuable experiences for the overall process:

- SVMs provide a suitable data classification model in CA scenarios of small and sparse training data.
- Sparse training data can be augmented in a semi-supervised classification scenario by features inferred from unsupervised topic models to improve classification quality.
- If machine classification is mainly targeted towards estimation on category proportions and trends in diachronic corpora instead of

classifying individual documents, already moderate performance on precision and recall of the classifier provides sufficient quality.

- Collection of training data in CA studies is expensive. It can be supported efficiently by processes of active learning, where analysts start with a small set of manually collected training data and iteratively augment this set by evaluating on examples suggested by a machine classifier.

- Selection of training examples for active learning randomly from a pool of data instances above a certain probability threshold for the positive category provides the best strategy to obtain a training set which validly identifies trends in time series data.

- Collecting around 400 training examples for a certain category or repeating active learning for at least eight iterations provides sufficient information to the classifier to estimate trends highly correlating with the actual data (Pearson's $r > 0.9$ for well-defined categories can be expected).

The workflow of classification for QDA was developed in this section on the basis of the MP data set as a kind of gold standard. It is applied in the next chapter together with the results of corpus exploration (see Section 3.2) to investigate on the discourse of democratic demarcation in Germany. For this, time series of several content analytic categories are computed and inspected in the document collection retrieved by the earlier IR process (see Section 3.1).