arXiv:2103.13550v1 [cs.CL] 25 Mar 2021

Term-community-based topic detection with variable resolution

Preprint

Andreas Hamm
Think Tank
German Aerospace Center DLR
Cologne, Germany
andreas.hamm@dlr.de

Simon Odrowski
Think Tank
German Aerospace Center DLR
Cologne, Germany
simon.odrowski@dlr.de

March 26, 2021

Abstract

Network-based procedures for topic detection in huge text collections offer an intuitive alternative to probabilistic topic models. We present in detail a method that is especially designed with the requirements of domain experts in mind. Like similar methods, it employs community detection in term co-occurrence graphs, but it is enhanced by including a resolution parameter that can be used for changing the targeted topic granularity. We also establish a term ranking and use semantic word-embedding for presenting term communities in a way that facilitates their interpretation.

We demonstrate the application of our method with a widely used corpus of general news articles and show the results of detailed social-sciences expert evaluations of detected topics at various resolutions. A comparison with topics detected by Latent Dirichlet Allocation is also included. Finally, we discuss factors that influence topic interpretation.

Keywords Text mining · Natural language processing · Topic modeling · Term ranking · Community detection · Corpus analysis · Word embeddings

1 Introduction

Facing an ever-growing amount of text data, automated methods of text evaluation have become indispensable for finding and analyzing information. Computerized information retrieval started to evolve many decades ago as one of the earliest fields of computer applications but continues to make spectacular progress in the context of recent machine learning developments.

The classical information retrieval task is to serve some information need formulated as a concrete query. However, given the sheer volume of texts available, there are many situations where, before asking detailed questions, one must first gain some insight into what kind of information is contained in the texts at all and what subject areas are covered.

This is where automatic topic detection, also called topic mining or topic modeling, can help. This process takes a text corpus, i.e., a large collection of text documents, as input and produces as output a set of *topics* which are meant to represent the various subjects written about in the corpus documents. The identification of topics within a corpus can be used in many ways: for a quick overview of the content and a better understanding if little is known about the corpus or its context; for ordering the documents of the corpus, similar to a classification, but more flexible in that it allows one document to be assigned to several topics rather than belonging only to one class; for analyzing the temporal evolution of thematic content or its relation to metadata like authorship or publisher. Computationally, it also can be seen as a method of dimensionality reduction for the corpus documents and as such lends itself as a building block in further machine learning analyses of the documents.

Like with any computerized application, at both ends of the process some translation step is needed: at the input side a quantification which turns the corpus into some mathematical data structure, and at the output side an interpretation of what the algorithmically derived output actually means. The latter step can be highly problematic in situations involving natural language as it carries more ambiguity and context dependency than numerical or highly formalized data. This can be an obstacle for finding indisputable and verifiable interpretations. Therefore, involving subject experts who ideally are well trained in methods of text interpretation is crucial.

While this paper focuses on general and technical aspects of the method rather than on comprehensive domain applications, it is written as a collaboration of a computational data scientist and a political scientist in order to keep a good balance between the computational and the interpretive aspects. Political science is, in fact, one of the domains that benefit most from reliable methods for automated topic discovery: While text is an indispensable source of knowledge about politics, the discipline, in line with the general trend, has recently been confronted with a flood of relevant textual material [1, 2, 3]. The background which motivated the present research is the need to scan and understand the strategic significance of huge amounts of incoming text documents of scientific, political, social and economic nature in a strategic unit of a large research organization.

Regarding the quantification of the corpus, the natural candidate for a data structure is the so called word-document matrix which keeps record of which words of the total corpus are contained in which document and how important they are for the document. The earliest approaches to topic discovery, going back to the 1980-ies, applied purely algebraic considerations to that word-document matrix. The currently predominant approaches to topic discovery can be grouped into two distinctly different lines: One is based on probabilistic generative models where topics are parameters (more specifically: probability distributions on words) that can be determined by statistical inference. The other one is based on transforming the word-document matrix into a network in which the topics show up as communities of strongly linked nodes. We will mention references for some of the existing variants of both lines in the next section.

There is a striking imbalance between the popularity of the two lines. The number of research papers using probabilistic topic models exceeds the number of publications following network-based approaches of topic detection by two orders of magnitude. However, in spite of many impressive successful applications of the probabilistic models it is not at all clear that they offer the best solutions in all situations. An initial investigation of several network-based methods in our group [4] showed a very promising potential and motivated further improvements of network-based topic detection which will be described in this paper.

The structure of the paper is as follows: In the next section we give a brief overview of some of the related work in the areas of topic mining and community detection. Section 3 presents our particular version of network-based topic detection. We exemplify the method by applying it to a well known corpus of BBC news reports [5] which has been widely used for text classification, topic modeling, and other text mining tasks in the literature (e.g., [6, 7]). In Section 4 we investigate the influence of two of the adjustable parameters of the method: the reduction percentage and the resolution parameter, and show how the latter one can be used to identify more and more topics on finer scale. For comparison, in Section 5 we apply the best-known probabilistic topic modeling approach, Latent Dirichlet Allocation (LDA), to our example corpus and elaborate on observations regarding topic interpretability and other differences. Section 6 draws some conclusions.

There are three main new contributions of this paper to the field of topic discovery: First, we describe a particular method for term ranking which is an important ingredient for producing and interpreting high-quality topics; second, we employ for the first time in the context of term co-occurrence networks the Leiden algorithm for a generalized modularity optimization as a tool for topic detection and show that the resolution parameter of the generalized modularity can be used for controlling the resulting topic granularity, which is particularly relevant from a domain expert perspective; third, we present new insight into questions of topic interpretability on the basis of expert evaluations and supported by word embeddings.

2 Related work

Methods of automatic topic detection (as well as other methods used in this article: keyword extraction and word embeddings) are based on the *distributional hypothesis* [8], which states that observations about the distribution of word occurrences allow to draw conclusions about semantics. The first systematic approach to topic detection was *Latent Semantic Indexing* (LSI) [9], based on singular value decomposition of the word-document matrix. Another successful algebraic method uses *non-negative matrix factorization* (NMF) [10].

Going beyond purely algebraic operations, *probabilistic Latent Semantic Analysis* (pLSA) [11] regards the observed word distribution in the documents as the outcome of a stochastic process that results from the mixture of two multinomial distributions, which can be reconstructed using stochastic inference. Latent Dirichlet Allocation (LDA) [12] follows a similar strategy but goes one step further in assuming that the mixture is not between fixed but random multinomial distributions, which are drawn from a Dirichlet distribution.

LDA has become enormously popular, not least because of several easy-to-use software implementations which employ efficient inference techniques like *collapsed Gibbs sampling* [13]. It has been applied to many diverse text collections like scientific publications, news collections, literary corpora, political debates, historical documents, social media posts, and many others; for reviews we refer to [14, 15, 16]. On the other hand, LDA motivated the development of a plethora of similar generative models with the aim of improving the method or of taking better account of special properties of the text collections to be studied. An example are generative models which can detect hierarchies of topics [17, 18].

Probabilistic topic models can be further enhanced by supplementing the word co-occurrence information with document metadata, like information on authorship, geographical location or relatedness to events [19, 20]. The Structural Topic Models (STM), which have proven useful in political science applications, also belong to that category [21].

A fundamentally different line of topic detection methods arose from graph-theoretical evaluation of word-document co-occurrences. We refer to [22] for a survey of graph-based text analysis. So called co-word maps were produced in a semi-manual fashion in early bibliometric studies [23]. TopCat [24] is one of the first graph-based procedures of topic detection. It is based on hypergraph clustering in co-occurrence hypergraphs of so called frequent-itemsets of named entities. Another approach is known under the name KeyGraph. It started as a method for key word extraction [25] based on a word co-occurrence graph on sentence level, but was later extended for event detection [26, 27].

Increased interest in network analysis and in particular the concept of community detection furthered the development of graph-based topic discovery. We refer to [28, 29] for reviews on community detection. Several methods of community detection have been used for topic discovery: [30] finds topics as communities in a KeyGraph by the Girvan-Newman algorithm involving the edge-betweenness [31]. Instead, [32] employs modularity maximization [33], using the Louvain algorithm [34]; similar approaches can be found in [35, 36, 37, 38]. Louvain-based community detection was also applied in [39, 40], in combination with a principle component analysis, to co-word maps. The Infomap algorithm [41] for community detection via a random walk was used in [42]. [43] identifies topics as cliques in a word co-occurence network. The hierarchical semantic graph model in [44] is based on a hierarchy of terms and uses subgraph segmentation via the normalized cut algorithm [45] for community detection. [46] finds topics as communities in a bipartite document-word graph with a Stochastic Block Model [47]. This approach establishes an interesting connection to the probabilistic topic models, as a Stochastic Block Model itself is a generative model. In fact, this graph-based topic detection method is closely related to pLSA. On the other hand, Stochastic Block Models also have been shown to be related to maximing [48] a parametrized generalized modularity [49].

In passing we remark that network analysis is applied to document collections not only in the form of word co-occurrence networks, but also by studying co-author and citation networks, and both, too, have been exploited for topic discovery [50, 51].

Term ranking will play an important role in our approach. On document level, term ranking is closely related to the problem of unsupervised key word extraction. This field is summarized in [52, 53]. On corpus level, we are not aware of any method that is comparable to ours. However, there is a vague resemblance to the method for detecting *hot words* in microblogs described in [54].

Word embeddings like Word2Vec [55] are a very efficient way of capturing the contextual information contained in large text collections for use in semantic text analysis. While we use pre-trained fastText embeddings [56] for structuring and assessing the topics that we find by detection of term communities, other authors have used similar word embeddings directly for identifying topics through clustering in the embedding space [57] or indirectly for improving probabilistic topic models [58].

3 A term-community-based topic discovery method

It is a very intuitive idea that topics within a text corpus show up as patterns in the overall word usage in the corpus documents. Graphs as the mathematical structure for representing networks of entities (so called *nodes* or *vertices*) which are linked (through so called *edges*) are an obvious choice for formalizing this idea. What is less obvious is which of the many possible ways of transforming the corpus into a graph is the most effective one for the present purpose and how exactly topic-forming patterns can be identified.

In this section we describe our particular choice which we found to be successful in the analyses of many text corpora. We give reasons for several specifics in our procedures but do not claim that we have tested all or even a big fraction of the conceivable alternatives. Optimizing model parameters is particularly problematic in the present situation not only because of the extension of the parameter space but also because of the lack of a convincing target function which describes the model success. We will come back to this point in Section 5.

As mentioned before, we use a corpus of 2225 news articles from BBC news dating back to 2004 and 2005 [5] as an example corpus for explaining and evaluating our method. The documents of this corpus each consist of a few hundred words.

3.1 Setting up the corpus network

Based on a corpus $\mathcal{D} = \{d_1, ..., d_N\}$ of N text documents d_i, we define the corpus graph as a weighted undirected graph $G_p(\mathcal{D}) = (V_p(\mathcal{D}), E_p(\mathcal{D}), W_p(\mathcal{D}))$, consisting of a vertex set, an edge set, and an edge weight function.

The vertex set $V_p(\mathcal{D})$ is a certain subset of unique and normalized words or word combinations appearing in the corpus (which hereafter we will subsume under the name *terms*), where p with $0 < p \leq 100$ is a parameter that controls the number of vertices.

More specifically, for $p = 100$ the terms that form the vertex set $V_{100}(\mathcal{D})$ result from a fairly standard text preparation pipeline, consisting of tokenization, lemmatization (in order to consolidate several inflected forms of the same word), and the removal of stopwords (frequent words with little meaning), short tokens (less than three characters), exotic characters, and tokens consisting mainly of digits. We also retain only nouns, adjectives, and proper nouns as usual in NLP tasks that focus on factual content.

Applying the procedures listed in the previous paragraph to the original documents leads to single-word term vertices only. Yet, retaining compound terms that consist of several words as units in the corpus graph is a desirable enhancement of the method because it prevents the loss of meaning that is a consequence of splitting up a compound term into its components. Technically, we can include compound terms without changing the pipeline described above using a formal trick: Before we put the documents into that pipeline, we connect the individual words of compound terms by underscores (e.g., “department of homeland security” becomes “department_of_homeland_security”). This renders the compound term a single token which survives the pipeline as a unit. However, identifying compound terms in documents is not an easy task. We experimented with various statistical and linguistic approaches. While it is possible to identify many meaningful combinations in that way, one also produces several nonsensical combinations which can create serious confusion in the results. Thus, we decided not to search for general compound terms but only for those which show up in named entity recognition. Concretely, we incorporate entities of the types events, facilities, geographical and political entities, languages, law, locations, nationalities or religious or political groups, organizations, persons, products, and works of art which consist of 2, 3, or 4 words.

For all these document preparation steps, we use the Python library spaCy [59]. Working with the small language model en_core_web_sm turned out to be sufficient for our purposes; using larger language models did not lead to significant changes in the results.

After these preparations every document d_i has been stripped down to a collection of terms which still carry the main subject content of the document. $V_{100}(\mathcal{D})$ is the set of all unique terms remaining in the corpus. However, usually quite a few of these terms are of general nature and not important for the main message of the document. Having those terms in the corpus graph blurs its ability to represent thematic links. Working with a smaller subset of $V_{100}(\mathcal{D})$, which we denote as $V_p(\mathcal{D})$ where $p < 100$ is the percentage of terms retained, can prevent that effect as we will discuss in more detail in Section 4.

In order to judge about which terms to drop from the corpus graphs, a function for *document term ranking* is needed which produces a rank order of terms depending on their significance for the message of the document. This is a well-known task in the context of key term extraction. The naïve solution would be to rank a term t in document d by its frequency $\mathrm{Tf}_d(t)$ in the document but it is well-known that this unjustly favors terms that tend to occur frequently, independent of the specific content of the document. The long-standing solution to this problem is to counterbalance the effect of overall frequent words by the *inverse document frequency*,

$$\mathrm{Idf}(t) = \log \frac{N}{1 + \mathrm{Df}(t)}$$

where $\mathrm{Df}(t)$ is the number of corpus documents that contain the term t, and to use $\mathrm{Tf}_d(t) * \mathrm{Idf}(t)$ for term ranking in the document d.

However, this typical bag-of-word approach—where all words of a document are treated equally independent of their position—neglects the observation that important words of a document are usually not distributed evenly over the whole document. Rather, they tend to appear in groups, and for many document types it is also common that authors place especially many words that characterize the document content at the top of the document, notably in title, subtitle, and abstract. A term ranking method that takes these two observations into consideration is PositionRank [60], which is a modification of the TextRank method introduced in [61] in analogy to the PageRank [62] method for ranking within a network of linked web pages.

In order to combine the advantages of the frequency arguments and the positional arguments for term ranking, we devised our own ranking method, posIdfRank [63], which works as follows: For a corpus document d, define a graph $g(d) = (v(d), e(d))$ (not to be confused with the corpus-wide graph $G_p(\mathcal{D})$) which has the set of unique terms of d as vertex set $v(d)$, and an edge $\epsilon_{st} \in e(d)$ between two vertices s and t iff the shortest distance between the terms s and t within the document d is less than or equal to some window size w (which is an adjustable parameter of the method).

We now consider a random walk on $g(d)$ in which the transition probability between two terms s and t is given by

$$P_{st} = \alpha \frac{\mathrm{Idf}(t) f_{st}}{\sum_{u \in v(d)} \mathrm{Idf}(u) f_{su}} + (1-\alpha) \frac{(1+\mathrm{pos}(t))^{\beta}\, \mathrm{Idf}(t)}{\sum_{u \in v(d)} (1+\mathrm{pos}(u))^{\beta}\, \mathrm{Idf}(u)}$$

where f_{st} counts how often in d the terms s and t have a distance less than or equal to the window size w, and $\mathrm{pos}(t)$ counts on which position within the document the term t appears first; α $(0 \leq \alpha \leq 1)$ and β $(\beta < 0)$ are two more parameters of the method. This process mimics a reader randomly scanning the document for important terms: From term s he moves with probability α to another term t in the vicinity (neighborhood of size w)—more likely reaching at terms which commonly stand close to s and which do not appear in many documents of the corpus. But with probability (1-α) he jumps to some word which can be far away from s—then more likely to terms at the beginning of the document (where the preference of terms at the beginning is stronger the smaller one chooses β) and again to terms which do not appear in many documents of the corpus.

The long term behavior of this random walk is characterized by its stationary distribution, a probability distribution $r_d(t)$ on $v(d)$. We regard this function as a useful document term ranking function: It has high values at terms that are frequently visited during the random walk.

For the calculations in this paper, we fix the three parameters of the method at $\alpha = 0.85$, $\beta = -0.9$ and $w = 5$. These values were derived from a best fit with the manually assigned key words of the dataset of journal abstracts from [64], which also showed that the ranking is only weakly sensitive to moderate changes of the parameter values so that it is not critical that other training datasets result in slightly different optimal parameter values.

We illustrate the considerations on document term ranking with the following example document from the BBC corpus:

> India widens access to telecoms
> India has raised the limit for foreign direct investment in telecoms companies from 49% to 74%.
>
> Communications Minister Dayanidhi Maran said that there is a need to fund the fast-growing mobile market. The government hopes to increase the number of mobile users from 95 million to between 200 and 250 million by 2007. "We need at least $20bn (£10.6bn) in investment and part of this has to come as foreign direct investment," said Mr Maran. The decision to raise the limit for foreign investors faced considerable opposition from the communist parties, which give crucial support to the coalition headed by Prime Minister Manmohan Singh. Potential foreign investors will however need government approval before they increase their stake beyond 49%, Mr Maran said. Key positions, such as those of chief executive, chief technology officer and chief financial officer are to be held by Indians, he added.
>
> Analysts and investors have welcomed the government decision. "It is a positive development for carriers and the investment community, looking to take a longer-term view of the huge growth in the Indian telecoms market," said Gartner's principal analyst Kobita Desai. "The FDI relaxation coupled with rapid local market growth could really ignite interest in the Indian telecommunication industry," added Ernst and Young's Sanjay Mehta. Investment bank Morgan Stanley has forecast that India's mobile market is likely to grow by about 40% a year until 2007. The Indian mobile market is currently dominated by four companies, Bharti Televentures which has allied itself with Singapore Telecom, Essar which is linked with Hong Kong-based Hutchison Whampoa, the Sterling group and the Tata group.

If one would simply go by the frequency of terms in the document, the 10 highest ranked terms would be:

> investment, market, foreign, mobile, India, telecom, government, investor, chief, indian

This list obviously does contain useful key terms of the document, but also very unspecific terms like "market" and "chief". In contrast, the top ranking according to $\mathrm{Tf}_d(t) * \mathrm{Idf}(t)$ would be:

> Maran, investment, telecom, mobile, indian, foreign, India, investor, market, direct

Here, the unspecific terms disappear or get shifted to lower positions. The very specific person name Maran, in contrast, appears at the top of the list.

Finally, the ranking according to posIdfRank results in the following top terms:

India, telecom, investment, foreign, limit, Maran, direct, investor, mobile, Sanjay_Mehta

This list now favors important terms of the title and subtitle, and also brings up person names which stand close to other key terms in the document. Among the three ordered term lists, this is the one which condenses the content of the document best.

Now that we have a document term ranking function we can define the corpus vertex set $V_p(\mathcal{D})$: It is the result of keeping only the top p percent (in the sense of the document term ranking function posIdfRank) of unique terms of each document.

In order to demonstrate the effect of this document reduction, we show what remains from the above example document if we keep only terms from $V_{50}(\mathcal{D})$:

India access telecom India limit foreign direct investment telecom Communications Minister mobile market government mobile investment foreign direct investment Maran decision limit foreign investor considerable opposition communist crucial coalition Minister Manmohan_Singh foreign investor government Maran investor government decision investment indian telecom market Gartner principal Kobita_Desai relaxation market indian telecommunication Ernst Young Sanjay_Mehta investment bank Morgan_Stanley India mobile market indian mobile market bharti_televenture Essar hutchison_whampoa Sterling

Having established the vertex set, we now have to specify the edge set $E_p(\mathcal{D})$ and the edge weight function $W_p(\mathcal{D})$. The edges are supposed to connect terms that are likely to help in identifying thematic relationships, and the edge weight function should measure how significant this connection is. Among all conceivable options, the simplest turns out to be very effective already, namely that two terms are connected if they appear together in at least one document and the weight of that connection is the number of documents in which both terms co-occur:

$$e_{st} \in E_p(\mathcal{D}) \iff (s, t \in E_p(\mathcal{D}) \wedge \exists i \in \{1, \dots, N\} : s, t \in d_i)$$
$$W_p(\mathcal{D})(e_{st}) = \#\{i \in \{1, \dots, N\} : s, t \in d_i\}$$

Several other authors who use network-based topic detection work with more restrictive and complicated edge definitions and weights, involving thresholds and Idf-values in order to avoid noise produced by accidental or meaningless co-occurrences. We did not find this beneficial in our case as we avoided such type of noise already by reducing the vertex set $V_p(\mathcal{D})$ with percentages p well below 100.

We will go into more detail concerning the influence of the value of p in section 4. In the present section we work with $p = 50$.

More concretely, following the procedures described above, the BBC corpus of 2225 documents leads to a graph $G_{50}(\mathcal{D}_{\mathrm{BBC}})$ that has 23859 vertices and 1896807 edges. The edge weights vary between 1 and 86 (with highest weight on the link between the terms *election* and *Labour*, meaning that 86 documents of the corpus contained both these terms).

3.2 Detecting topics as communities

The aim of setting up the graph $G_p(\mathcal{D})$ was that in it topics of $\mathcal{D}$ would show up as communities of terms, where we refer to the network-theoretical concept of communities, which are—loosely speaking—groups of nodes that are densely connected within each group and sparsely connected with other groups. While this vague idea is quite intuitive, there are various non-equivalent options how to turn it into a quantifiable criterion for identifying communities, some of which we have mentioned in Section 2. It is a priori not clear which criterion fits best for the language related purpose of establishing dense thematic connections. Comparative work in [4] indicates that modularity maximization is very well suited for achieving the goal of topic detection.

Given an undirected weighted graph $G = (V, E, W)$, modularity is a function which maps a partition $\mathcal{C} = \{C_j, j = 1, \dots, k\}$ of V into vertex groups $C_j \subset V$ (with $\dot\bigcup_{j=1}^{k} C_j = V$) to a real number which measures how well the groups C_j can be considered to be communities in the sense of high intra-group connectedness. It can be thought of as consisting of two parts. Denoting for a vertex $s \in V$ the group of $\mathcal{C}$ to which s belongs by $c(s)$ and defining the total edge weight $M = \frac{1}{2} \sum_{s,t \in V} W(e_{st})$ the first part,

$$\mathcal{I}(\mathcal{C}) = \frac{1}{2M} \sum_{s,t \in V} W(e_{st}) \delta_{c(s)c(t)},$$

represents the fraction of all intra-group edge weights compared to the total edge weight of G, where we have used Kronecker's δ notation in order to express that s and t should be in the same group. The second part is the same fraction, but not for the graph at hand, G, but what one would expect for a random graph that has the same degree distribution (i.e., edge weight sum connected to each vertex) as G:

$$\mathcal{J}(\mathcal{C}) = \frac{1}{(2M)^2} \sum_{s,t \in V} K_s K_t \delta_{c(s)c(t)},$$

with $K_s = \sum_{t \in V} W(e_{st})$, the edge weight sum (degree) at vertex s.

Modularity now is the difference between these two terms:

$$\mathcal{Q}_1(\mathcal{C}) = \mathcal{I}(\mathcal{C}) - \mathcal{J}(\mathcal{C}).$$

Maximizing the modularity therefore means finding a partition of the vertex set such that within the groups of this partition the fraction of intra-group edges is as big as possible when compared to the fraction of intra-group edges which one would expect in a similar random graph. At least intuitively, this translates well into what we are looking for: communities of terms that appear more often together in common documents than one would expect if the terms were randomly distributed.

The above definition of modularity can be generalized to include a parameter γ like follows:

$$\mathcal{Q}_\gamma(\mathcal{C}) = \mathcal{I}(\mathcal{C}) - \gamma\mathcal{J}(\mathcal{C}).$$

Maximizing $\mathcal{Q}_\gamma(\mathcal{C})$ leads to coarser or finer communities depending on the value of γ: In the extreme situation that $\gamma = 0$, the objective function is just $\mathcal{I}(\mathcal{C})$, and its maximum is obviously reached at the trivial solution $\mathcal{C} = \{V\}$, i.e., when the whole graph is considered to be one big community. In the other extreme, $\gamma \to \infty$, the objective is to minimize $\mathcal{J}(\mathcal{C})$, and this obviously happens when $\mathcal{C} = \{\{t\} : t \in V\}$, i.e., when each single vertex forms its own community such that the number of communities is equal to the number of vertices.

Varying γ in the generalized modularity $\mathcal{Q}_\gamma(\mathcal{C})$ makes it possible to find communities of variable granularity, and therefore γ is called the resolution parameter.

Computationally, maximizing the (generalized) modularity is known to be a non-deterministic polynomial-time hard problem, which means that there are no efficient algorithms that guarantee an optimal solution. However, several efficient heuristic algorithms are known for producing potentially suboptimal but useful solutions.

Here, we use the recently published Leiden algorithm [65], which is an improvement of the very popular so called Louvain algorithm [34] for maximizing modularity. Starting from the extreme partition $\{\{t\} : t \in V\}$ where each vertex forms its own community, the Leiden algorithm first visits the nodes in a random order and tries greedily to shift nodes to other communities in a way that offers the biggest modularity increases. In a second step (which is the main improvement compared to the Louvain algorithm), the community partition is refined in a way that produces well-connected communities. After that, an aggregated graph is formed which contains the refined communities of the original graph as vertices. The whole procedure is repeated for the aggregated graph, and this is iterated until no further modularity increase can be achieved.

For our calculations we use the implementation of the Leiden algorithm in the Python version of the library igraph [66].

Applying the Leiden algorithm with the standard resolution parameter $\gamma = 1$ to the example graph $G_{50}(\mathcal{D}_{\text{BBC}})$ results in 8 to 14 term communities, depending on the run. Each run may produce a different number of communities because the algorithm follows a non-deterministic heuristics. Therefore, not only the number but also the terms of the communities, which are only approximations to the optimal community partitioning, may change in each run. However, closer inspection shows that the differences between the runs are not large. In particular, if a run produces more than 8 communities, the additional communities are strikingly smaller than the biggest 8 communities. This is an indication that in some runs the greedy algorithm fails to assign some terms to any of the dominant communities and leaves them in some small residual communities.

While these odd cases are reasonably easy to detect, it is nevertheless better to remove those undecisive terms completely from the picture. Therefore, for the actual topic detection we use 20 runs and consider only those sets of terms which end up together in the same Leiden community in at least 15 of the runs. Finally, of those sets we retain only the ones that contain at least 10% of the terms that one would expect in a topic if all topics were of equal size. In this way, the remaining sets of terms form stable topics.

Following this procedure, 22426 terms among the 23859 vertices of $G_{50}(\mathcal{D}_{\text{BBC}})$ (or 94%) can be assigned to one of 8 stable topics. The biggest of these topics contains 4657 terms, the smalles 214 terms.

Based on the assignment of terms to topics, we can also determine which document of the corpus is concerned with which topics. The connection is made by counting the relative number of topic terms within a document: If $f_i(d)$ is the number of terms in document d that belong to the topic cluster C_i, then $\frac{f_i(d)}{\sum_{j=1}^{k} f_j(d)}$ is a good indicator of the importance of topic i for that document. While one can argue that here a probabilistic topic model would offer a sounder method for calculating the topic share per document, we found that a simple count of topic terms works very well as we will show when we use the term community topics for document classification in Section 4.

Like with other topic detection approaches, the method results in a list of terms that characterize a topic, but the actual interpretation of what that topic is about, is left to human evaluation. It is certainly a difficult task to look at hundreds or thousands of terms, having to make sense of what topic might be encoded in them. Probabilistic topic models that produce topics as probability distributions on terms have an advantage here at first glance: The topic terms can be sorted by their probability, and one usually looks only at the 10 to 30 most probable terms.

In the next subsection we will explain how we suggest to look at big sets of topic terms in a way that facilitates interpretation in the absence of a probability distribution on the terms.

3.3 Topic presentation

Looking in a random order at the hundreds or thousand terms which constitute a topic is certainly not helpful for grasping its meaning. It would be best to look at the most characteristic terms first; what we need is a term ranking. In the context of network preparation we have already established a *document term ranking* $r_d(t)$. However, what is required now is a *corpus term ranking*. The former one can only decide which terms are the most characteristic ones for a certain document d, but now we need a more global ranking function $r(t)$, independent of d. For a term t, the document term ranking function $r_d(t)$ is only defined for those documents d which contain t. Since the document frequency $\mathrm{Df}(t)$ varies a lot with t it would not be fair to simply take the arithmetic average of the existing values of $r_d(t)$. This resembles star ratings in recommender systems where typically some items have been rated many times and some items have only one rating. There one uses a Bayesian average for ranking [67].

In order to transfer that solution to the problem of corpus term ranking, we first introduce a discretized version $q_d(t)$ of the document term ranking $r_d(t)$. This is because we do not claim that the document term ranking is sufficiently exact to measure the importance of terms continuously but rather that it is a good base for grouping terms into sets of more or less important terms. Let $T(d) = (t_1, \ldots, t_{m_d})$ be an ordered sequence of the terms in document D, $t_i \in d$ for $i = 1, \ldots, m_d$ and $r_d(t_i) > r_d(t_j)$ if $i < j$. Then we divide $T(d)$ into A parts $T_a(d)$, $a = 1, \ldots, A$, of equal lengths $[m_d/A]$. We also introduce a cut-off value $K \leq A$; A and K are parameters which can be adjusted so as to result in a ranking that works well.

$$q_d(t) = b \iff t \in T_{K+1-b}(d) \quad \text{for} \quad b = 1, \ldots, K$$

and $q_d(t) = 0$ otherwise. After some experimentation we fixed $A = 20$ and $K = 3$; this means that the top 5% of terms in a document get the discretized rating 3, the next 5% the rating 2, and the third 5% the rating 1.

Based on $q_d(t)$, we calculate the corpus term ranking function $r(t)$ as the following Bayesian average:

$$r(t) = \frac{C * L + \sum_{d:t\in d} q_d(t)}{C + \mathrm{Df}(t)}$$

with $C = \frac{\sum_{s\in V_{100}(\mathcal{D})} \mathrm{Df}(s)}{\#(V_{100}(\mathcal{D}))}$, which is the document frequency averaged over all terms, and $L = \frac{K(K+1)}{2A}$, which is the mean $q_d(t)$ ranking of all terms.

With the function $r(t)$ we can sort all terms of the corpus.

The ten terms which ranked highest as the most specific terms for the corpus $\mathcal{D}_{\mathrm{BBC}}$ are:

Yukos, Holmes, UKIP, Fiat, Blunkett, howard, Yugansk, Kenteris, Parmalat, Wenger

The ten terms ranked lowest—the least specific ones—are:

year, place, time, month, spokesman, recent, week, Tuesday, long, second

Now we can present the terms that form a topic in the order of decreasing values of $r(t)$. The examples in Table 1 show how advantageous this is. Both columns show terms from the same topic found by community detection in $G_{50}(\mathcal{D}_{\mathrm{BBC}})$ with resolution parameter $\gamma = 1$. The topic comprises a total of 4080 terms. The left column of the table shows just 18 randomly picked topic terms. They do not give a clear indication about the meaning of the topic. In contrast, the right column shows the 18 topic terms with highest values of $r(t)$. They clearly give a much better impression of what the topic is about.

Table 1: Comparison of selected terms of the same topic. The left column picks some random terms; the right column shows the terms with highest r value.

random order	r-sorted order
contribution	UKIP
supporter	Blunkett
Czechoslovakia	howard
unwinnable	Silk
sentencer	Kilroy
Skidelsky	Speaker
tract	Guantanamo
registration	Kennedy
Julian_Lewis	Hague
flexible	hunting
elizabeth_blackman	Blair
Lawless_World	blair
feeling	quango
Eady	Mr_Blunkett
Moazzam	Guantanamo_Bay
secretary	lib_dem
Brady	migrant
understanding	Goldsmith

However, looking at only one or two dozens of 4080 identified topic terms wastes a lot of information, and we suggest to look at many more terms when interpreting topics in order to achieve a proper assessment. In order to facilitate an overview over many topic terms, next to the specificity ranking $r(t)$ we add another structuring criterion to the set of topic terms by grouping them into clusters of semantically related terms. Here, we make use of pretrained fastText embeddings [56]. The fastText approach belongs to the semantic work embedding methods through which words can be mapped to a moderately low-dimensional Euclidean vectorspace in a way that semantic closeness translates into small metric distances. The pre-trained fastText models are shallow neural nets with output vectors of dimension 300 that were trained on huge text collections from Wikipedia and Common Crawl, using the so called CBOW task of predicting a word by its surrounding words. In distinction to its predecessor Word2Vec, fastText internally does not work on the level of words but on the level of its constituting character n-grams. In the present context this offers two advantages: First, this mapping works even on words which do not appear in the Wikipedia and Common Crawl collections; second, word variations due to spelling mistakes or imperfect lemmatization usually end up close to each other in the vector space representation.

If we now take the vector representations of the topic terms, we can use any metric-based clustering method for finding groups of semantically related words, or, more precisely, of words whose components have been seen frequently together in huge text collections. After some experimentation, we decided to use hierarchical clustering in its scikit-learn implementation AgglomerativeClustering with distance threshold 1.

We show these semantic groups of terms, which we call semantic strata, rather than single terms, when we present the topics. We order the strata by the value of $\max r(t)$ for the terms t per stratum. As a result we have a two-dimensional order in the topic terms: one dimension ranking the specificity and one dimension depicting semantic relations.

For topic evaluation we produce large sheets with the topic terms structured in the stratified way described above. In Table 2 we show only the beginning of such a sheet for better comprehension. The rows depict the strata in which the r-sorted top terms of the topic get accompanied by semantically related terms, or in the case of person names by persons who usually appear in a common context.

Table 2: Principle of stratified topic view. Terms with high r values are complemented through other topic terms from their fastText embedding cluster.

UKIP			
Blunkett	Mandelson	Miliband	Heseltine
howard			
Silk			
Kilroy			
Speaker			
Guantanamo	detainee	prisoner	captivity
Kennedy	Moynihan		
Hague			
hunting	pheasant	Hunting	
Blair	Blairs		
blair			
quango	Quangos		
Mr_Blunkett	David_Blunkett	David_Miliband	david_miliband
Guantanamo_Bay	mr_hague	the_british_national_party	the_world_economic_forum
lib_dem	White_Paper	Royal_Mail	Upper_House
migrant	refugee		

4 The influence of the resolution parameter and of the reduction percentage

In this and the following section, we present concrete observations derived from working with the BBC corpus. The general aim is to study the influence of modeling decisions and parameters on structure, interpretability, and applicability of the detected topics.

All topic term sheets produced in the way described in Section 3, with various values of the parameters γ and p discussed in the following, were given to three evaluators from the social sciences with varying backgrounds in political science, economics, and sociology. Each topic was assessed independently by two of them: The task was to interpret the topic and to find an appropriate label for it. The evaluators also graded the interpretability of the topic from score 1 (= hardly possible to make sense of the topic) to score 5 (= topic recognizable without any doubts). The third evaluator compared the two previous evaluations and prepared a meta evaluation. In the majority of cases, the two first evaluators gave identical or nearly identical labels, which were almost always approved by the meta evaluator. In most other cases, it was possible to agree on a consensus label; in very few cases, the evaluators found the topics unidentifiable. The average of the three evaluator scores was given as final score.

4.1 Varying the resolution

Next, we study the influence of the resolution parameter γ at the example of the BBC corpus, rank-reduced to 50% of its terms.

Table 3 shows the resulting number k of topics for different values of γ; some of which were chosen for later comparison in Section 5. As expected, k rises with γ. The table also shows that the evaluators generally rated topic interpretability, expressed through the average score $\bar{z}$, as high. However, there is a clear trend that interpretability declines with higher resolutions.

Up to resolution $\gamma = 1.5$, there are hardly any topics that were difficult to interpret, but at $\gamma = 2.5$, 15% of the topics were found to be problematic. Nevertheless, it is remarkable that the method succeeds in producing $k^+ = 67$ clearly interpretable topics.

We will come back to issues of interpretability in Section 5 but first discuss content aspects of increasing the resolution. We start with looking closer at the topics at resolution $\gamma = 0.8$.

The BBC corpus comes with a classification into 5 broad classes: Business, Entertainment, Politics, Sport, Tech. Resolution $\gamma = 0.8$ produces 5 topics, labeled by the evaluators as follows: Sports, Music & films, Technology, Politics

Table 3: Evaluation of topics produced from $G_{50}(\mathcal{D}_{\mathrm{BBC}})$ with various values of the resolution parameter γ. The table shows for each resolution the number k of topics produced. The numbers k^+ and k^- indicate how many topics the evaluators found easy to interpret (score 5 or 4) and hard to interpret (score 1 or 2), respectively; $\bar{z}$ is the mean score of all topics of that resolution.

γ	k	k^+	k^-	$\bar{z}$
0.80	5	5	0	5.0
1.00	8	8	0	4.8
1.07	10	10	0	4.8
1.37	19	18	0	4.7
1.50	27	25	1	4.5
2.00	58	44	6	4.2
2.50	89	67	14	3.8

(UK interests based), Economy. The congruence between classes and topics is obvious on the level of labels. In order to see how well this extends to the document level, we calculate the topic shares in each document. The topic with the highest share we call the dominant topic of the document. In this way we compile the crosstable Table 4 between preassigned classes and detected dominant topics.

Table 4: Crosstable between preassigned classes and detected dominant topics ($\gamma = 0.8$) for all documents of the BBC corpus

Topic Class	Economy	Music & films	Politics	Sports	Technology
Business	473	4	18	3	12
Entertainment	7	346	12	3	18
Politics	18	1	396	0	2
Sport	1	1	1	507	1
Tech	5	4	11	3	378

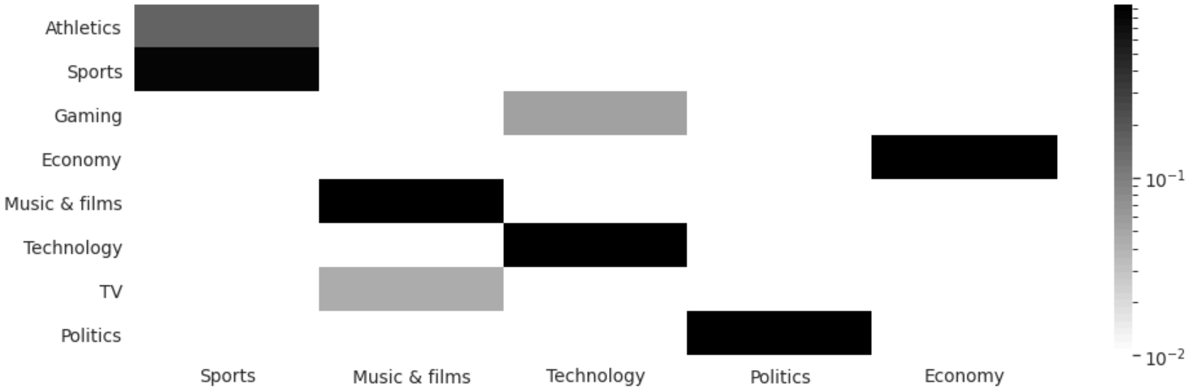
The corresponding classification statistics is shown in Table 5. However, what we have in the detected topic distribution is more than a simple classifier, as we do not only learn which is the dominant topic of a document but also what other topics are visible in a document. For instance, the example document about foreign investment in Indian telecom presented in Section 3 belongs to the class Business. But in the topic term assignment, while Economy is the dominant topic with a share of 59%, there is also a share of 25% Politics and of 14% Technology, which is a reasonable topic composition for the document. In fact, several of the few cases which were misclassified based on their dominant topic were borderline articles with two similarly strong appropriate topics. The following titles give some examples: The article “News Corp eyes video games market” belongs to class Business, but Technology was detected as dominant topic. The Entertainment article “Ethnic producers face barriers” has Politics as dominant topic. The article “Arsenal may seek full share listing” is in class Business, but the topic Sports dominated here.

Table 5: Classification statistics for predicting preassigned classes by detected dominant topics ($\gamma = 0.8$). Precision is the fraction of true positives among all positive predictions, recall is the fraction of true positives compared to all actual class members, and the f1-score is their harmonic mean. Values are given for each class separately and as an average weighted according to the sizes of the classes.

	precision	recall	f1-score
Business	0.938	0.927	0.933
Entertainment	0.972	0.896	0.933
Politics	0.904	0.950	0.926
Sport	0.983	0.992	0.987
Tech	0.920	0.943	0.931
weighted avg	0.945	0.944	0.944

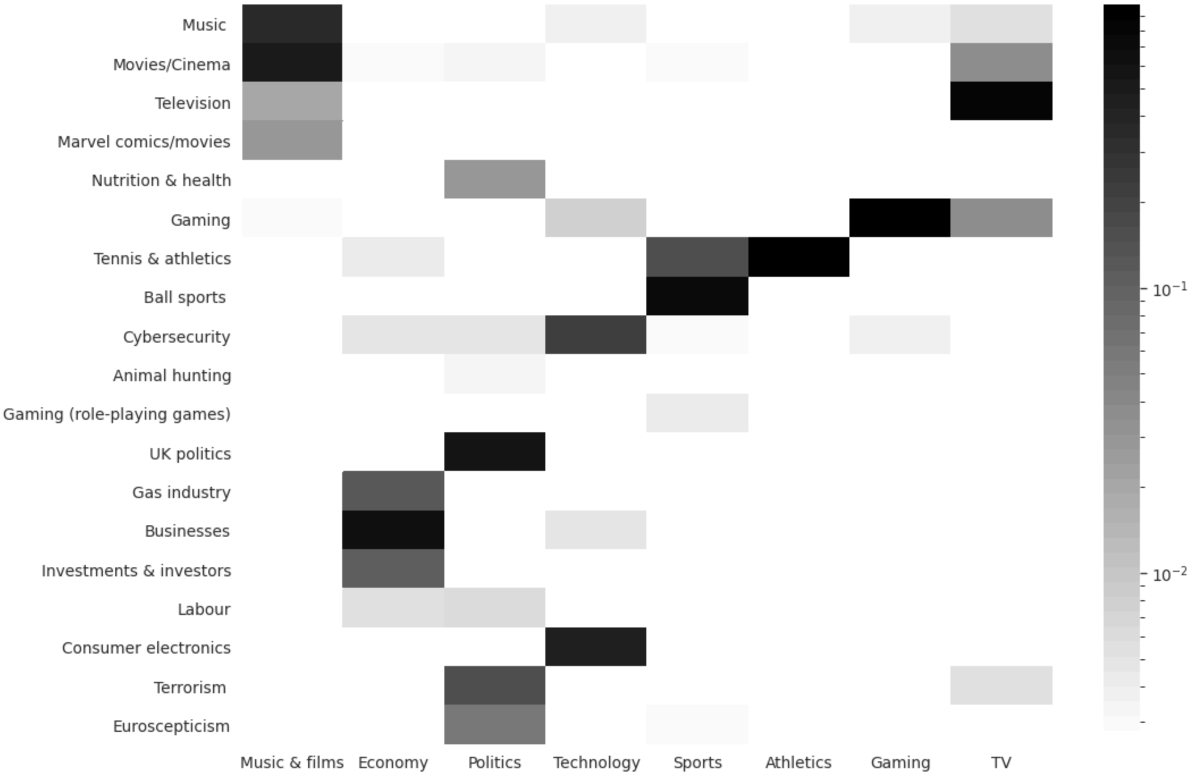
We are now interested in what topics show up when we increase the resolution parameter γ. Resolution $\gamma = 1$ results in 8 topics. The heat map in Figure 1 shows how the topic terms of these 8 topics—corresponding to the 8 rows—are distributed within the 5 topic of lower resolution $\gamma = 0.8$—corresponding to the 5 columns. While the 5 original topics basically persist, 3 of them have smaller spin-offs: The general Sports topic gave rise to a new Athletics topic, from the Music & films topic a new TV topic splits off, and the Technology topic forks into a new (Video) Gaming topic.

Figure 1: Heat map showing how topic terms of the 5 $\gamma = 0.8$ topics (columns) get distributed on 8 topics at $\gamma = 1.0$ (rows)



This phenomenon of small topics splitting off from a big one is a typical pattern contributing to the increasing number of topics at higher values of γ. The comparison of resolution $\gamma = 1.37$ with 19 topics and $\gamma = 1.0$ with 8 topics in Figure 2 shows further examples. The topic Politics decomposes into the big topic UK politics and small topics Terrorism, Euroscepticism, and Nutrition & health. However, two more topics have significant contributions from the former Politics topic: Labour, which has also input from the former Economy topic, and Cybersecurity, which is primarily fed by the former topic Technology.

Figure 2: Heat map showing how topic terms of the 8 $\gamma = 1.0$ topics (columns) get distributed on 19 topics at $\gamma = 1.37$ (rows)



Another example illustrates that direct topic division is not the only way in which topics of higher resolution emanate. The high-resolution topics Music, Movies/cinema, Television, Marvel comics/movies altogether arise from a joint restructuring of the two coarser topics Music & films and TV (and some faint influences of other topics).

Yet another form of recombination can be seen in the class Sport. At $\gamma = 1$, it comprises two topics: Sports and Athletics. At $\gamma = 1.37$, there are still only two sport topics: Ball sports (with an emphasis on football and rugby) and a combined topic Tennis & athletics. This means that here the higher resolution splits off Tennis from the general Sports topic but immediately combines it with the Athletics topic.

Going from $\gamma = 1.37$ to $\gamma = 1.5$, we end up with 27 topics (see Figure 3). Here, remarkable developments are that the business and investment topics give now rise to 4 related topics, 2 of them now getting into regional or sector details (Development Asia and Aviation). Also the sport topics get more specific: Football and Rugby are separate topics now.

Continuing the route to $\gamma = 2.0$ and $\gamma = 2.5$, further specific topics can be detected: the separation of Tennis and Athletics, various country specific topics, additional industry sectors like the Automotive industry (see Table 6). All in all, this confirms the role of γ as resolution parameter. However, as topics that are not easy to interpret become more frequent (see Table 3), one cannot reach arbitrarily high resolution.

Figure 3: Heat map showing how topic terms of the 19 $\gamma = 1.37$ topics (columns) get distributed on 27 topics at $\gamma = 1.5$ (rows)

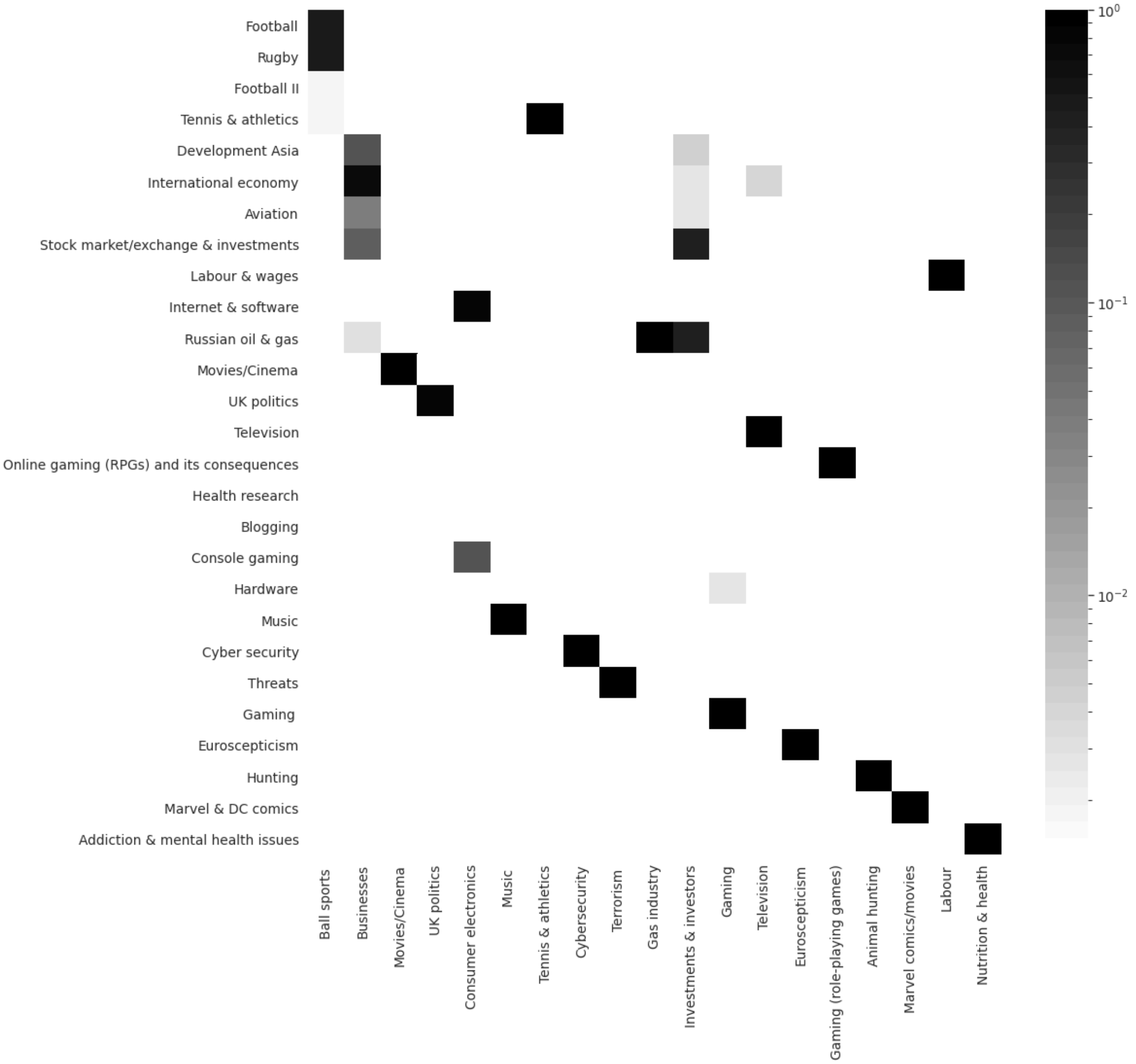


Table 6: Beginning of the stratified topic sheet for one of the 89 topics at $\gamma = 2.5$ which can be interpreted as Automotive. The terms in this clipping are dominated by car makes (LVMH being an outlier, though one can imagine why it is included in the same cluster). While lower parts of the sheet not visible here also contain general terms from the subject field, such as "motorsport", "sportscar", "racing", "braking", "steering", "throttle", and "gearbox", the specificity ranking puts the makes into higher position.

Fiat			
general_motor	Aston_Martin	mitsubishi_motor	Budget_Aston
grid			
Rover			
LVMH			
Peugeot			
Mini			
Ferrari	Mercedes	Maserati	
factory	Sindelfingen	Dingolfing	Ruesselsheim
Mitsubishi	Chrysler	DaimlerChrysler	vehicle
Cadillacs	Cadillac	limousine	
saab			

4.2 The effect of term reduction

In all of the above examples, we have used $G_{50}(\mathcal{D}_{\mathrm{BBC}})$, i.e., the BBC corpus with $p = 50\%$ of its ranked terms. Here, we briefly investigate what happens when we work without reduction ($p = 100\%$), or if we reduce much more ($p = 10\%$).

With reference to Table 7, we discuss the effect on topic number and topic interpretability. At low resolution ($\gamma = 1.0$), we observe that increasing reduction (decreasing p) gives rise to a larger topic number. This is easily understandable as in the term community approach common terms are the glue which binds the community. Having more but less specific terms supports the formation of bigger clusters. On the other hand, smaller numbers of terms make big clusters unlikely and rather produce larger numbers of smaller topics.

Table 7: Evaluation of topics produced from $G_p(\mathcal{D}_{\mathrm{BBC}})$ with resolution parameters $\gamma = 1.0$ and $\gamma = 2.0$ at three different values of the reduction percentage p. The table shows for each case the number k of topics produced. The numbers k^+ and k^- indicate how many topics the evaluators found easy to interpret (score 5 or 4) and hard to interpret (score 1 or 2), respectively; $\bar{z}$ is the mean score of the topics.

p	γ	k	k^+	k^-	$\bar{z}$
100	1.0	5	5	0	5.0
100	2.0	98	48	33	3.2
50	1.0	8	8	0	4.8
50	2.0	58	44	6	4.2
10	1.0	38	18	11	3.1
10	2.0	100	41	39	3.0

While the lack of terms clearly is problematic for the topic interpretability at $\gamma = 1.0$ and $p = 10$, the case $\gamma = 1.0$ and $p = 100$ works well as it reproduces the 5-class structure of the corpus again—like for $\gamma = 0.8$ and $p = 50$. In fact, if the only purpose of topic detection was classification with respect to the coarse 5 classes of the corpus, the choice $\gamma = 1.0$ and $p = 100$ would be preferable, as a comparison of Tables 8 and 9 with Tables 4 and 5, respectively, shows. However, the unreduced $p = 100$ wins its slightly better ability to predict the dominant topic at the cost of recognizing secondary topics less well.

At higher resolution, $\gamma = 2.0$, the advantages of a moderate reduction of topic terms are clearly visible: $p = 100$ and $p = 10$ produce a comparable number of well interpretable topics as $p = 50$, but both extreme choices also yield many uninterpretable topics. These typically consist of few topic terms. In the case of $p = 100$, many unspecific terms are involved, whereas in the case of $p = 10$ the topic terms are so narrow that they seem to belong to one document only.

Table 8: Crosstable between preassigned classes and detected dominant topics ($p = 100$, $\gamma = 1.0$) for all documents of the BBC corpus

Topic Class	Economy	Entertainment	Politics	Sports	Technology
Business	491	1	12	0	6
Entertainment	8	354	12	0	12
Politics	12	0	402	2	1
Sport	2	0	2	507	0
Tech	12	7	6	6	370

Table 9: Classification statistics for predicting preassigned classes by detected dominant topics ($p = 100$, $\gamma = 1.0$); to be compared with Table 5.

	precision	recall	f1-score
Business	0.935	0.963	0.949
Entertainment	0.978	0.917	0.947
Politics	0.926	0.964	0.945
Sport	0.984	0.992	0.988
Tech	0.951	0.923	0.937
weighted avg	0.955	0.955	0.955

Term reduction is about finding the right balance between removing as many unspecific terms as possible and keeping enough terms for characterizing topics in detail. While there is no practicable way to predict the optimal value of p, we can provide some guidelines, based on the results presented here as well as on findings from further tests and work with different text corpora. These include summaries of scientific studies [68] and parliamentary documents [69], but also abstracts of scientific articles, RSS news feeds, and mixed corpora. More generally, the longer the documents are, the stronger the reduction (i.e., the lower p) should be. More specifically, for articles with one or two pages, reductions between 50% and 25% work—by and large equally—well. Short texts like abstracts work better with less reduction, whereas for long documents with many pages reduction between 10% and 20% is helpful, also as it decreases the term network size and consequently the computational effort.

5 Topic interpretability and comparison with LDA

The term community method for topic detection described above is geared towards good topic interpretability. The results presented in Section 4 confirm that evaluators found the topics uncovered by the method to be of high quality in that sense. Before we look in more detail into the factors that determine the topic interpretability we apply standard Latent Dirichlet Allocation to the same corpus for the sake of comparison.

5.1 LDA topics for the BBC corpus

LDA is based on the assumption of a generative process where topics are considered to be probability distributions over all words of the corpus, which are not directly observable but latent in the documents in that each document is a random mixture of topics. The word distributions within the topics as well as the topic distributions within the documents are assumed to have the form of a Dirichlet distribution. Fixing the number k of topics within the corpus and further hyper parameters that determine the shape of the Dirichlet distribution, methods of statistical inference can be used to determine the latent distributions of words per topic from the observed distributions of words per document. In particular, the method of Gibbs sampling is known to produce convincing topics in many applications.

This motivates a comparison of the topics detected as term communities with the topics identified using LDA. For generating these topics, we used the popular Mallet LDA toolkit [70] through the Python wrapper contained in the library Gensim [71]. We fixed the topic number k to values described below and used the defaults for all other parameters. This means that further tuning might improve the results shown below, but also that we compare with the typical way in which LDA is used in the applied literature.

The resulting topics were presented to the evaluators as lists of terms, sorted by their probability, where only topics with a probability greater than 0.001 were shown. The labeling and evaluation process was carried out in the same way as

Table 10: Evaluation of topics of the BBC corpus produced with a standard LDA procedure in comparison to term community topics in $G_{50}(\mathcal{D}_{\mathrm{BBC}})$. In each block, the table shows an LDA model and the corresponding term communities (TeCo) resulting in the same number of topics and lists for each model the number k of topics produced, the numbers k^+ and k^- indicating how many topics the evaluators found easy to interpret (score 5 or 4) and hard to interpret (score 1 or 2), respectively, and the mean score $\bar{z}$ of the topics. For LDA models the coherence value c_v is also given.

$Model$	k	k^+	k^-	$\bar{z}$	c_v
LDA $k = 5$	5	5	0	5.0	0.59
TeCo $\gamma = 0.8$	5	5	0	5.0	
LDA $k = 10$	10	9	0	4.4	0.62
TeCo $\gamma = 1.07$	10	10	0	4.8	
LDA $k = 19$	19	13	1	4.1	0.61
TeCo $\gamma = 1.37$	19	18	0	4.7	
LDA $k = 27$	27	20	4	4.0	0.57
TeCo $\gamma = 1.5$	27	25	1	4.5	
LDA $k = 58$	58	39	9	3.7	0.57
TeCo $\gamma = 2.0$	58	44	6	4.2	

for the term communities. Table 10 shows how the evaluators graded the topics for 5 different values of k. For reasons of comparison we also include rows from Table 3 for term communities that resulted in the same number of topics. Altogether, the table shows that the evaluators found most LDA topics well interpretable, but the scores are consistently below the results for the term community method.

The evaluators report that cognitive processing of stratified term clusters (with a significant proportion of named entities), as they were presented in the case of term communities, appears prima facie more complex than that of word lists (with more general terms), as in LDA. After familiarizing with both ways of presentation through interpreting a couple of topics, however, more information at a glance and more details eventually increase the interpretability of topics. In particular, this results in fewer unidentifiable topics and better differentiability between topics. Furthermore, while interpreting topics from both methods becomes more difficult with increasing topic numbers, this effect is stronger for LDA.

Conversely, stratified word clusters including a decent amount of named entities were considered very fruitful from a domain application perspective: Generally speaking, more, and more detailed, information (actors, issues, places, time references, aspects) about the topics and the corpus as such is beneficial for most social science purposes, and hardly ever an impediment.

The clearest results are obtained at $k = 5$, corresponding to the number of preassigned classes. This model can be evaluated as a classifier in the same way we did for term community topics at $p = 50, \gamma = 0.8$ and at $p = 100, \gamma = 1.0$. The results are presented in Tables 11 and 12. Precision and recall are high, but clearly below the values for the term community method.

Table 11: Crosstable between preassigned classes and detected dominant topics according to LDA with $k = 5$ for all documents of the BBC corpus; to be compared with Tables 4 and 8

Topic Class	Economy	Music & films	Politics	Sports	Technology
Business	482	4	14	0	10
Entertainment	14	355	7	1	9
Politics	40	8	358	4	7
Sport	15	38	14	444	0
Tech	15	12	2	24	348

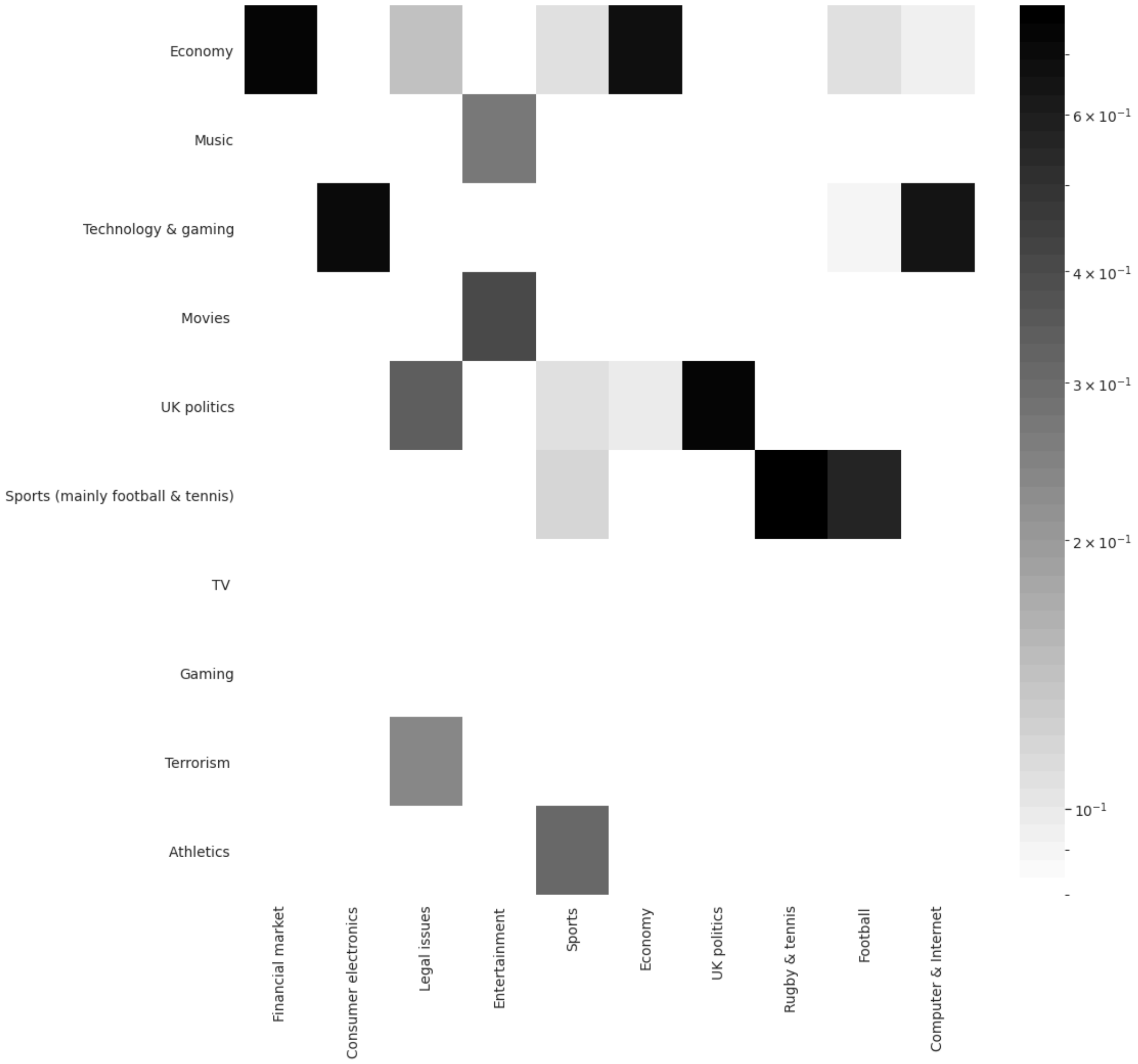
In fact, with typical LDA reasoning one would argue that one should not work with $k = 5$ at all but rather tune k such as to achieve an optimal result. Very often optimization of LDA hyperparameters is understood to be targeted at maximizing coherence of the topics. According to [72], one of several coherence measures, usually called c_v, is especially well correlated with human evaluation of coherence. It is calculated from the co-occurrence statistics of topic

Table 12: Classification statistics for predicting preassigned classes by detected dominant topics from LDA with $k = 5$; to be compared with Tables 5 and 9

	precision	recall	f1-score
Business	0.852	0.945	0.896
Entertainment	0.851	0.920	0.884
Politics	0.906	0.859	0.882
Sport	0.939	0.869	0.902
Tech	0.930	0.868	0.898
weighted avg	0.896	0.893	0.893

words within text windows in the corpus documents. Table 10 shows this value for the various LDA models. In the present situation, c_v reaches its maximum at $k = 10$ with another local maximum at $k = 19$; $k = 5$ and the two higher k values are clearly suboptimal with respect to c_v. Our evaluators' scores are not convincingly correlated to c_v.

Figure 4: Heat map showing how topic terms of the $k = 10$ LDA topics (columns) get distributed on 10 term community topics at $\gamma = 1.07$ (rows)



It is tempting to draw from the results presented so far the conclusion that LDA and term communities offer two options for finding the same topics, both normally working well, with slight advantages concerning interpretability on the side of the term communities. Instead, further analysis of the term composition of the topics shows that both methods find different topics. This can be seen in the heat map in Figure 4. The columns show the LDA topics found at $k = 10$, the rows display the term communities for $P = 50, \gamma = 1.07$. The heat map grayscale indicates how many of the most probable terms of an LDA topic come from which term community. If there was a 1-to-1 correspondence of topics, the heat map would only show exactly one single dark square for every row and every column. However, this is not the case. Rather, there are LDA topics that extend over several term communities (e.g., Entertainment comprising Music and Movies), and there are term communities that contain several LDA topics (e.g., Technology & (video) gaming containing Consumer electronics and Computer & Internet). In LDA, Sports is a very broad topic, Financial market is a very narrow topic. In term communities, Economy and UK politics are very broad, while the communities for TV and for (Video) Gaming are so small that they do not contain any of the most likely LDA terms.

Altogether, LDA and term communities seem to offer complementary views of the subjects discussed in the corpus.

5.2 Factors influencing the topic interpretability

Based on the evaluation of the interpretability of many topics produced with two totally different methods of topic evaluation, we want to look into factors which make the difference between topics that can be recognized easily and topics that defy interpretation. Therefore, we asked the evaluators in each case of a topic with low score what caused the difficulties with interpretation. One obvious reason which effected both—topics presented as term communities as well as LDA topic term distributions—was that topic terms seemed to point into contradictory directions. However, there were two other equally important reasons: First, some topics consisted of only a few dozen terms without projecting a clear picture about a common theme—this was a relatively frequent problem for the term community method; second, there were topics that showed almost exclusively generic terms, which lacked expressiveness and made it impossible to recognize a specific theme—this is a common problem for LDA. These two effects can be traced back to a common root: the lack of informative terms. To put it differently: A topic does not only need non-contradictory but also informative terms in order to be easily interpretable.

The corpus term ranking which we established in Section 3 offers a way to separate informative from non-informative terms—also when talking about terms in the LDA term distributions: We consider a term only as informative if its ranking value is higher than a certain threshold. For the BBC corpus it works well to define the set of highly informative topic terms of topic C as $H = \{t \in C : r(t) > 0.25\}$. In order to assess the degree of informative terms in a topic, we simply count $M_H = \#H$.

Assessing the risk of contradictions within the highly informative terms can be done by some form of coherence measure. Rather than using a self-referential intrinsic measure that compares to co-occurrences within the corpus documents or working with a larger external corpus, we suggest to use a word embedding coherence as defined in [73] which we base here on the same fastText embedding that we have used already in Section 3, as it derives relatedness of terms from an extremely large text collection:

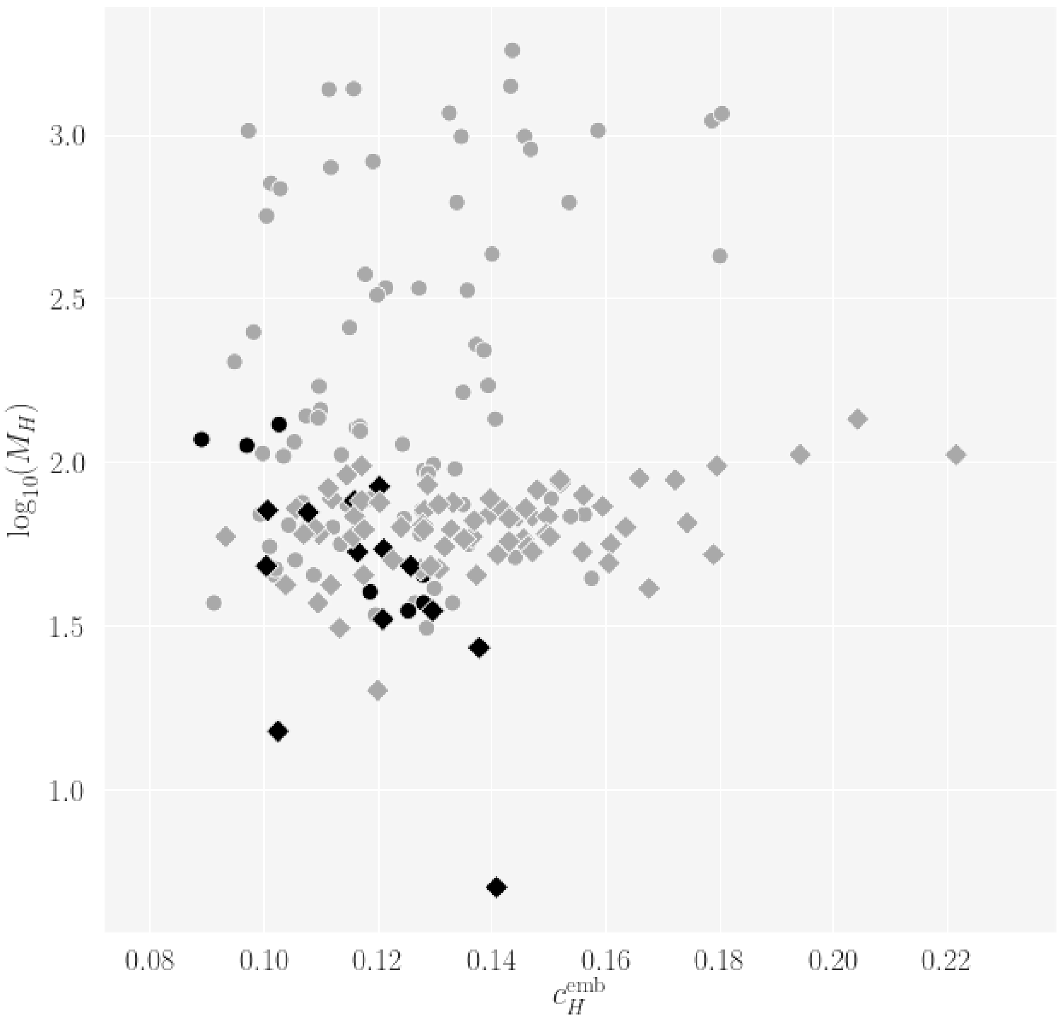
$$c_H^{\mathrm{emb}} = \frac{1}{M_H(M_H - 1)} \sum_{\substack{s,t \in H \\ s \neq t}} \mathrm{fTsim}(s, t)$$

where $\mathrm{fTsim}(s, t)$ is the cosine similarity between the two fastText vectors for s and t.

Figure 5 shows how topics which were evaluated as difficult for interpretation are positioned with respect to the two measures M_H (amount of informative terms) and c_H^{emb} (consistency of informative terms). Included are 85 topics from term community detection ($P = 50$ with $\gamma = 1.37$ and $\gamma = 1.5$) and 85 topics from LDA ($k = 27$ and $k = 58$). Topics with good interpretability are colored gray, the 20 topics for which the evaluators gave low scores are colored black. Term community topics are marked as circles, LDA topics as diamonds. Obviously, all problematic topics are gathered in the lower left quadrant of relatively small coherence and small term number, confirming the above hypothesis about the two factors that can cause difficulties for interpretation.

This also shows that the coherence measure alone is not a criterion which can predict topic interpretability, at least not when comparing different methods of topic detection. In fact, even a combination of coherence and term number cannot tell whether or not evaluators will find it easy to recognize the meaning of a topic. There are several examples in the left lower quadrant where topics were clear. But producing topics with many informative terms and considerable word embedding coherence does reduce the risk of ending up with meaningless topics.

Figure 5: Scatterplot visualizing a relation between topic interpretability, term specificity, and coherence. LDA topics are represented through diamonds, term community topics through circles. Topics that the evaluators scored with grade 3 or higher are colored gray; black color marks the topics that were found difficult to interpret. The topics are positioned vertically according to the logarithm of the number of its highly ranked terms M_H and horizontally according to the embedding coherence c_H^{emb} of the highly ranked terms.



6 Conclusions and outlook

Term community detection using parametrized modularity in a rank-reduced term co-occurrence network results in topics that are nearly always easy to interpret by domain experts. This observation is not only substantiated by the extensive studies on the BBC news corpus presented here but also by experience with numerous other corpora that our group monitors in the course of strategic analyses: political documents, scientific publications, government notifications, research news, general news. The ability to produce topics on different resolution levels by varying a continuous parameter is a feature that is particularly relevant from a domain expert perspective since text corpora are assumed to contain topics on several levels of granularity. For example, one might be interested in broader thematic areas (like health policy), discourses within these areas (like Corona/COVID-19), subjects within these discourses (like vaccinations), issues within these subjects (like vaccine prioritization) or even events/incidents within these issues (like deviations from prioritization plans in a particular vaccination center).

Our special form of term ranking on corpus level and additional clustering in word embedding space allow a comprehensive but lucid presentation of topic terms which supports topic interpretation. Along this line we have developed

TeCoMiner [68], a software tool for interactively investigating topics as term communities. There we take advantage of the fact that computing time for finding term communities is relatively small, faster than running a sufficient number of Gibbs sampling iterations for LDA.

Term community detection is one of many methods for discovering topics in large text collections. While it is natural to ask which of those methods works best, one can hardly expect a clear answer to this question: One reason is that no quantitative criterion is known that adequately predicts the human interpretability of a topic; as we have seen, topic coherence alone is certainly not sufficient. Another reason derives from the fact that certain methods are known to work better or worse depending on some properties of the corpus—a one-suits-all method might not exist. This is particularly true for generative topic models which always are based on quite specific and hardly verifiable assumptions about modeling details like the conditional dependence structure and prior distributions.

The phenomenological approach of term community detection however has worked reasonably well for all corpora we have tried and thus can be recommended for initial overview but also for deeper insight with higher resolution when investigating unknown corpora. Nevertheless, we saw when comparing with LDA topics that it certainly is interesting to run different methods on the same corpus as this may produce complementary topics.

The fact that different topic detection methods find different topics can be compared to a recent observation for community detection [74], where typically the communities also show considerable variation depending on the method used. There it was pointed out that all the differing communities can still be seen as different arrangements of the same building blocks. The same might be true for topics: They may be conceived of as arrangements (of elementary events, incidents, and concepts) that emerge with the corpus itself, rather than as pre-existing ideas from which the authors of documents could choose at the time of writing. The various topic detection methods search for such arrangements in different ways, thereby shedding light on the discourse underlying the corpus from distinct, but most likely complementary, perspectives.

Regarding future work, it will certainly be worthwhile amalgamating the network-theoretical approach to topic detection with other network-theoretical text mining procedures like co-authorship or citation networks, and with knowledge graphs.

The method presented here is likely to be particularly well suited for applications in the social sciences, especially as it produces both informative and well interpretable topics on different levels of thematic granularity. It should therefore be considered as a promising alternative to probabilistic topic modeling.

Acknowledgements

We are deeply grateful to Jana Thelen, whose work on a predecessor version of the method described here motivated and shaped several of its details. Special thanks go to Rasmus Beckmann and Mark Azzam for fruitful discussions and constant support. We also acknowledge the domain expert support of Friderike Uphoff and Stefan Odrowski in evaluating the topics.

References

[1] Burt L. Monroe and Philip A. Schrodt. Introduction to the special issue: The statistical analysis of political text. *Political Analysis*, 16(4):351–355, 2008.

[2] Henry E. Brady. The challenge of big data and data science. *Annual Review of Political Science*, 22(1):297–323, 2019.

[3] Ken Benoit. Text as data: An overview. In *The SAGE Handbook of Research Methods in Political Science and International Relations*, pages 461–497. SAGE Publications Ltd, 2020.

[4] Jana Thelen. Methoden der Netzwerkanalyse im Topic Modeling. Master's thesis, Department of Mathematics and Computer Science, University of Cologne, 2020. https://elib.dlr.de/141146/.

[5] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*. ACM Press, 2006.

[6] V.S. Anoop, S. Asharaf, and P. Deepak. Unsupervised concept hierarchy learning: A topic modeling guided approach. *Procedia Computer Science*, 89:386–394, 2016.

[7] Svetlana S. Bodrunova, Andrey V. Orekhov, Ivan S. Blekanov, Nikolay S. Lyudkevich, and Nikita A. Tarasov. Topic detection based on sentence embeddings and agglomerative clustering with markov moment. *Future Internet*, 12(9):144, 2020.

[8] Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.

[9] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[10] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03*. ACM Press, 2003.

[11] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*. ACM Press, 1999.

[12] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[13] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Supplement 1):5228–5235, 2004.

[14] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[15] Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296, 2017.

[16] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211, 2018.

[17] David M. Blei, Michael I. Jordan, Thomas L. Griffiths, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, page 17–24, Cambridge, MA, USA, 2003. MIT Press.

[18] Justin Grimmer. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, 18(1):1–35, 2010.

[19] Jianwen Wang, Xiaohua Hu, Xinhui Tu, and Tingting He. Author-conference topic-connection model for academic network search. In *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*. ACM Press, 2012.

[20] E. Yan, Y. Ding, S. Milojevic, and C. Sugimoto. Topics in dynamic research communities: An exploratory study for the field of information retrieval. *J. Informetrics*, 6:140–153, 2012.

[21] Margaret E. Roberts, Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082, 2014.

[22] S. S.Sonawane and P. A. Kulkarni. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96(19):1–8, 2014.

[23] A. Rip and J. P. Courtial. Co-word maps of biotechnology: An example of cognitive scientometrics. *Scientometrics*, 6(6):381–400, 1984.

[24] Chris Clifton and Robert Cooley. TopCat: Data mining for topic identification in a text corpus. In *Principles of Data Mining and Knowledge Discovery*, pages 174–183. Springer Berlin Heidelberg, 1999.

[25] Y. Ohsawa, N. E. Benson, and M. Yachida. Keygraph: automatic indexing by co-occurrence graph based on building construction metaphor. In *Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries -ADL'98-*, pages 12–18, 1998.

[26] Yukio Ohsawa. KeyGraph: Visualized structure among event clusters. In *Chance Discovery*, pages 262–275. Springer Berlin Heidelberg, 2003.

[27] H. Wang, F. Xu, X. Hu, and Y. Ohsawa. Ideagraph: A graph-based algorithm of mining latent information for human cognition. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 952–957, 2013.

[28] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.

[29] Sanjay Kumar and Rahul Hanot. Community detection algorithms in complex networks: A survey. In *Communications in Computer and Information Science*, pages 202–215. Springer Singapore, 2021.

[30] Hassan Sayyadi and Louiqa Raschid. A graph analytical approach for topic detection. *ACM Transactions on Internet Technology*, 13(2):1–23, 2013.

[31] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[32] Shanliang Yang, Qi Sun, Huyong Zhou, Zhengjie Gong, Yangzhi Zhou, and Junhong Huang. A topic detection method based on KeyGraph and community partition. In *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence - ICCAI 2018*. ACM Press, 2018.

[33] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[34] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[35] Michael D. Salerno, Christine A. Tataru, and Michael R. Mallory. Word community allocation: Discovering latent topics via wordco-occurrence network structure. http://snap.stanford.edu/class/cs224w-2015/projects_2015/Word_Community_Allocation.pdf, 2015.

[36] Henrique F. de Arruda, Luciano da F. Costa, and Diego R. Amancio. Topic segmentation via community detection in complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(6):063120, 2016.

[37] Tommy Dang and Vinh The Nguyen. ComModeler: Topic Modeling Using Community Detection. In Christian Tominski and Tatiana von Landesberger, editors, *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2018.

[38] Minjun Kim and Hiroki Sayama. The power of communities: A text classification model with automated labeling process using network community detection. In *Proceedings of NetSci-X 2020: Sixth International Winter School and Conference on Network Science*, pages 231–243. Springer International Publishing, 2020.

[39] Loet Leydesdorff and Adina Nerghes. Co-word maps and topic modeling: A comparison using small and medium-sized corpora (n < 1,000). *Journal of the Association for Information Science and Technology*, 68(4):1024–1035, 2016.

[40] Tobias Hecking and Loet Leydesdorff. Can topic models be used in research evaluations? Reproducibility, validity, and reliability when compared with semantic maps. *Research Evaluation*, 28(3):263–272, 2019.

[41] M. Rosvall, D. Axelsson, and C. T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[42] Andrea Lancichinetti, M. Irmak Sirer, Jane X. Wang, Daniel Acuna, Konrad Körding, and Luís A. Nunes Amaral. High-reproducibility and high-accuracy method for automated topic classification. *Physical Review X*, 5(1), 2015.

[43] Wu Wang, Houquan Zhou, Kun He, and John E. Hopcroft. Learning latent topics from the word co-occurrence network. In *Communications in Computer and Information Science*, pages 18–30. Springer Singapore, 2017.

[44] Tingting Zhang, Baozhen Lee, Qinghua Zhu, Xi Han, and Edwin Mouda Ye. Multi-dimension topic mining based on hierarchical semantic graph model. *IEEE Access*, 8:64820–64835, 2020.

[45] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[46] Martin Gerlach, Tiago P. Peixoto, and Eduardo G. Altmann. A network approach to topic models. *Science Advances*, 4(7):eaaq1360, 2018.

[47] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, 2011.

[48] M. E. J. Newman. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5), 2016.

[49] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1), 2006.

[50] Jia Zeng, William K. Cheung, Chun hung Li, and Jiming Liu. Coauthor network topic models with application to expert finding. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, 2010.

[51] Zhen Guo, Zhongfei Zhang, Shenghuo Zhu, Yun Chi, and Yihong Gong. Knowledge discovery from citation networks. In *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009.

[52] Sifatullah Siddiqi and Aditi Sharan. Keyword and keyphrase extraction techniques: A literature review. *International Journal of Computer Applications*, 109(2):18–23, 2015.

[53] Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3):259–291, 2020.

[54] Zhiwen Yu, Zhitao Wang, Liming Chen, Bin Guo, and Wenjie Li. Featuring, detecting, and visualizing human sentiment in chinese micro-blog. *ACM Transactions on Knowledge Discovery from Data*, 10(4):1–23, 2016.

[55] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. https://arxiv.org/abs/1301.3781, 2013.

[56] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[57] Andrei M. Butnaru and Radu Tudor Ionescu. From image to text classification: A novel approach based on clustering word embeddings. *Procedia Computer Science*, 112:1783–1792, 2017.

[58] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2015.

[59] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[60] Corina Florescu and Cornelia Caragea. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017.

[61] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, 2004. Association for Computational Linguistics.

[62] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Networks*, 1998.

[63] Andreas Hamm. Complex word networks - comparing and combining information extraction methods. https://elib.dlr.de/127501/, May 2019. Contributed to SPCS2019, Stockholm.

[64] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, 2003.

[65] V. A. Traag, L. Waltman, and N. J. van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1), 2019.

[66] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research, 2006.

[67] Xiao Yang and Zhaoxin Zhang. Combining prestige and relevance ranking for personalized recommendation. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*. ACM Press, 2013.

[68] Andreas Hamm, Jana Thelen, Rasmus Beckmann, and Simon Odrowski. TeCoMiner: Topic discovery through termcommunity detection. https://arxiv.org/abs/2103.12882, 2021.

[69] Simon Odrowski and Andreas Hamm. Analyzing parliamentary questions: A political science application of a new topic modelling approach. https://elib.dlr.de/141131/, October 2020. Contributed to SocInfo 2020.

[70] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[71] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[72] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15*. ACM Press, 2015.

[73] Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. Using word embedding to evaluate the coherence of topics from twitter data. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*. ACM Press, 2016.

[74] Maria A. Riolo and M. E. J. Newman. Consistency of community structure in complex networks. *Physical Review E*, 101(5), 2020.