Reception of LGBT in Newspapers

Magdalena Bönisch Till Haubenreißer Maximilian Möller

February 27, 2017

Universität Leipzig, Introduction to Digital Humanities (Dr. Köntges)

Abstract Here are 150 to 200 words which constitute our abstract.

1 Introduction

This is our pretty Introduction.

2 Workflow & Implementation

In the following, the general workflow for analyzing the LGBT reception within newspapers is described. Subsequently, the concrete decisions concerning the implementation are addressed.

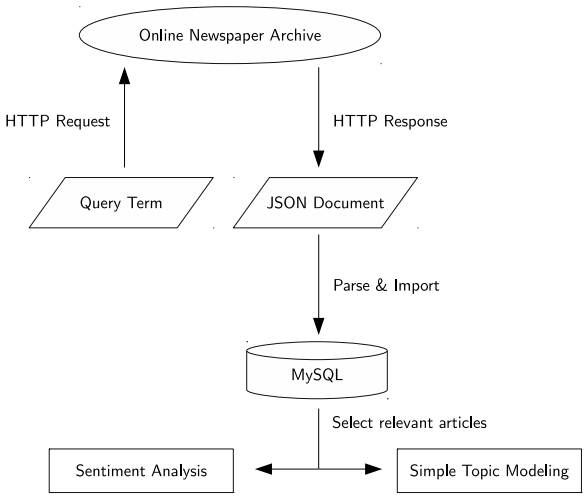

Figure 1: Workflow

Workflow

The overall workflow is shown in Fig. 1. The upper part reflects the first step of collecting newspaper articles and creating a database. For each term from a predefined list of query terms, an HTTP request to the API of an online newspaper archive is sent. In this work, the Article Search API of the New York Times[1] has been used as it provides access to articles published since its foundation in 1851 and thus enables an extensive time-dependent analysis. Furthermore, the returned JSON documents contain not only a URL to the actual article but also textual attributes like the lead paragraph and the publication date. The analysis is based on these text fields. Hence, no further call for the complete article is necessary. Since the access to the NYT API is limited per second and per day, the responses are parsed and stored into a MySQL[2] database. A relational database allows a quick access to the data along with SQL as a powerful query language for data selection and summarization. By using the database, subsequent analysis is based on the whole corpus independent of the API. Furthermore, the proposed workflow is more flexible with respect to adding data from an additional newspaper archive since its response only has to be mapped to the schema of the database while the analysis module (lower part of the workflow in Fig. 1) remains unaffected.

The analyses are based on two dimensions: query term and time. Thus, the input of the sentiment analysis and the simplified topic modeling approach is a set of lead paragraphs containing the query term and contained in articles published at a certain time. For instance, such an input can be all lead paragraphs from the database which contain the word *homosexual* and which were published in the 1990s.

The sentiment analysis assigns each sentence a vector of sentiment probabilities. We have considered five sentiments: *very negative*, *negative*, *neutral*, *positive*, and *very positive*. For example, the vector

$$v_s = \langle 0.2, 0.5, 0.1, 0.1, 0.1 \rangle$$

[1] `https://developer.nytimes.com/`
[2] `https://www.mysql.com/`

describes the case that a sentence s has a probability of 20% for having a very negative sentiment, 50% for having a negative sentiment, and 10% for having a neutral, positive, and very positive sentiment, respectively. For the topic modeling, the joint occurrence of a target term together with another word (cooccurrence) is analyzed. Target terms are LGBT-related words, like the query terms.[3] For each cooccurrence, its frequency and its significance is calculated. For instance, the cooccurrence

$$c_{2010s} = \langle \text{bisexual}, \text{rights}, 31, \text{true} \rangle$$

means that within the lead paragraphs from 2010 to 2019, *bisexual* and *rights* occurred 31-times in the same paragraph and that this occurrence is significant. By clustering significant cooccurrences, the topics were manually created in order to yield high-quality topics.

Implementation

The whole workflow has been implemented in Java as a Maven[4] project. The project is managed on GitHub.[5] It consists of one module for collecting and one for analyzing the data. The database is accessed by using the Java Database Connectivity (JDBC) API[6].

Collecting Module The used query terms are *bisexual*, *gay community*, *homosexual*, *lesbian*, *transgender*, and *transsexual*. However, the term *gay* was not considered as a query term because it appeared too often in non-LGBT-related contexts, as a given name or surname, for instance. In order to obey the time limit of the NYT API, only one request per second was sent. Almost all requests (95.4%) were successful. They returned a non-empty JSON object which was then inserted into the database. Failures of requests were due to denied access to the archive (HTTP 403) or gateway time-outs (HTTP 504). Because such failures were very rare, failed requests were not sent for a second time. All in all, the built corpus consists of 44,485 articles, 93,7% of which contain a non-empty lead paragraph on which the analyses are based.

The database models the mapping of query terms and keywords to articles. Keywords represent additional information on the articles such as associated persons, organizations, or geographical information. Approximately 75% of all articles have at least one keyword. Moreover, an article has a URL which points to its HTML representation in two-thirds of the cases; otherwise the article is only accessible as a PDF document.[7] Further attributes are the publication date, the actual text type (for instance, article, interview, or biography), the headline, an abstract, the lead paragraph, and a snippet. Since the headline is very short, the abstract is missing for 62% of the articles, and the snippet is mostly the same as the lead paragraph, the analysis is based on the lead paragraph. For the body of the article, the HTML representation of the article would have had to be requested and parsed.

Analysis Module Essentially, the analysis module of the Maven project comprises the sentiment and the topic model package. They both rely on the sentence extraction task. For a certain target term and a certain publication date, this task selects windows of sentences from the paragraphs in the database. Given a window size of 2, for instance, in addition to the sentence containing the target term, the two directly preceding as well the two directly succeeding sentences are extracted.[8] For the sentiment analysis, we have chosen a window size of 0, i.e. only the containing sentence, and for the topic modeling a size of 1. Both analyses depend on the natural language processing library Stanford CoreNLP[9]. This library contains annotators for tokenization, sentence splitting, part-of-speech tagging, and sentiment analysis. Since the sentiment annotator is based on the sentence structure, it is expected that it returns better results than approaches which only count the occurrences of particular negatively or positively connotated words or phrases while ignoring their syntactic context [SPW+13]. However, determining the sentence sentiment is a very time consuming operation. Therefore, we have chosen the minimal window size.

The simplified topic modeling approach consists of three steps: preprocessing, cooccurrence counting, and the statistical evaluation. The preprocessing is executed for the corpus only once. For all paragraphs,

[3] The query terms are used for building an LGBT-corpus. But within this corpus it is possible to search for other terms than query terms as well. For instance, *lesbian* is a query term as well as a target term. *gay* is only a target term.

[4] `https://maven.apache.org/`

[5] `https://github.com/macksimiljan/lgbt-news`

[6] `http://www.oracle.com/technetwork/java/javase/jdbc/index.html`

[7] Older articles are archived as PDF not as HTML.

[8] For the given corpus, a window size greater than 4 will have no difference to a smaller size since the paragraphs consists only of a few sentences. For instance, not more than 40% of the paragraphs consists of two or more sentences.

[9] `http://stanfordnlp.github.io/CoreNLP/`

the contained words and their number of occurrence are determined. Stop words and numbers are excluded from this word statistics. The stop word list is based on [MS03, p.533] whereas numbers are recognized by a regular expression. After having created the word statistics, it is possible to define the list of context words. A context word is a meaningful word which cooccurs with a target term. A word is assumed to be meaningful if it is not a stop word or a number and has a minimum frequency of 3. The frequency condition enforces that words being typographical errors are excluded from the analysis. After preprocessing, the word statistics consists of approximately $66 \cdot 10^3$ word types and $1.2 \cdot 10^6$ word tokens.[10] 43.4% of all words occur only once or twice. Thus, there are approximately $29 \cdot 10^3$ context words.

In the next step, all cooccurences of a target word w_{target} with one of the n context words are counted in the paragraphs containing w_{target} and published in a certain time span t. To yield the cooccurrences and their number, these paragraphs were tokenized and cleansed by removing all non-context words from the sentences and transforming the remaining words to lower case. For instance, the sentences

The president argues for gay rights. He is tolerant.

is mapped to

president argues gay rights tolerant

This sequence of words is used to built a context vector $v_t(w_{\text{target}}) = \langle c_1, c_2, \ldots, c_n \rangle$ for each w_{target} and t such that c_i represents the (absolute) frequency of the cooccurrence of w_{target} with the context word at list position i. Because cooccurrence is not a reflexive relation between two words, $c_i = 0$ if the context word at position i equals w_{target}. In the example, assume that the list of context words L consists of six words:

$$L = [\text{argues, gay, president, rights, tolerant, usa}]$$

Depending on the chosen maximum distance $d_{\max}$, the context vector $v_t(\text{gay})$ for the target word *gay* is built. Let $d_{\max}$ be 2, then there can be not more than $d_{\max} - 1 = 2 - 1 = 1$ word between the target word and its cooccurrence partner. This yields the context vector

$$v_t^1(\text{gay}) = \langle 1, 0, 1, 1, 1, 0 \rangle$$

[10] Since we followed a simple approach, we did not run a lemmatizer. For comparison, the Oxford Dictionary counts about $200 \cdot 10^3$ (lemmatized) word types (inclusive stop words) in the English language [OD].

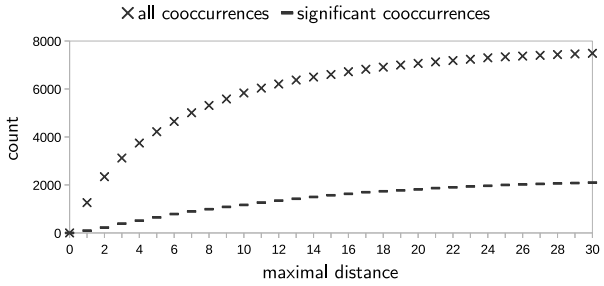

Figure 2: Convergence of the number of cooccurrences.

because every word except for *usa* cooccurs with *gay* exactly once. For $d_{\max} = 1$, only the direct neighbors of *gay* are considered resulting in the context vector

$$v_t^2(\text{gay}) = \langle 1, 0, 0, 1, 0, 0 \rangle.$$

Studies suggest that the average sentence length in written prose is between 20 and 25 tokens, for instance [Sic74]. The maximum distance $d_{\max}$ is not sentence-sensitive, i.e., it ignores sentence boundaries. Nevertheless, as the paragraphs are only a few sentences, there is a maximum distance $d'_{\max}$ such that for all distances greater than $d'_{\max}$ the number of cooccurrences is the same. Fig. 2 shows this for the count of cooccurrences with *gay* in the 2010s. The number of all cooccurrences as well as the number of significant cooccurrences converge to 7650 and to 2300, respectively. These values are reached for a maximum distance greater than 56 (not shown in the figure). Tests with further target words lead to similar results. However, the smaller the maximum distance, the more semantically connected the words of the cooccurrence are expected to be. Choosing a great distance, the cooccurrences can connect words which are probably in different sentences. Since the sentence boundary is a semantic boundary as well (each sentence represents a logical statement on its own), a great maximum distance results in cooccurrences which might include words that should not be interpreted as being semantically connected. Thus, we run the program with a maximum distance of 6 and of 14 for all target terms and times. For $d_{\max} = 6$ ($d_{\max} = 14$), approximately 67% (85%) of all cooccurrences and 40% (67%) of all significant cooccurrences are detected.

The last step is the calculation of significant cooccurrences. Following [Bor08, MS03], three significance measures were implemented: mutual information, log-likelihood, and t-score. For some test data, the t-score yields the best results. Let n be the number of context

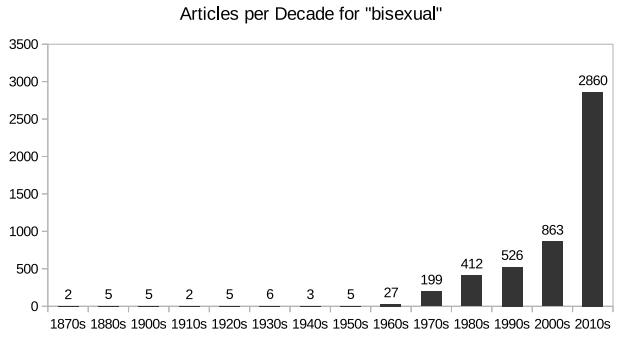

Figure 3: Number of articles for *bisexual*.

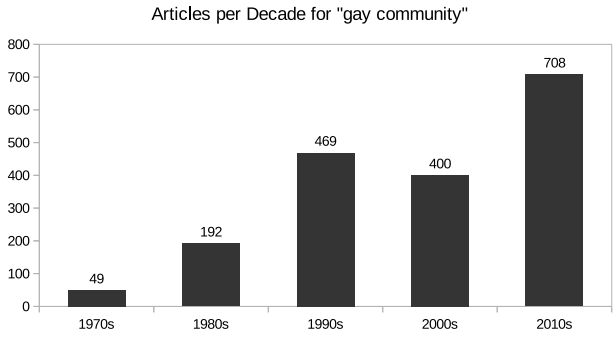

Figure 4: Number of articles for *gay community*.

words, n_{t} and n_{c} the number of occurrences of the target term and of a context word, respectively, and n_{tc} the count of the cooccurrence of w_{target} and w_{context}, then the t-score is defined according to Eq. 1.

$$\mathrm{t}(w_{\mathrm{target}}, w_{\mathrm{context}}) = \frac{n_{\mathrm{tc}} - \frac{n_{\mathrm{t}} \cdot n_{\mathrm{c}}}{n^2}}{\sqrt{n_{\mathrm{tc}}}} \qquad (1)$$

Consequently, a cooccurrence is defined as significant if it passes the t-test for $p = 0.005$. Additionally, all cooccurrences were selected occurring more often than on average in order to get further hints for topics.

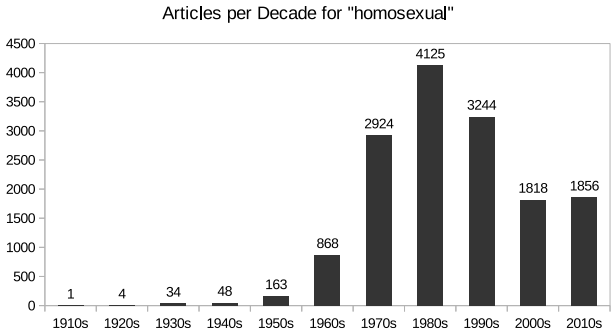

Figure 5: Number of articles for *homosexual*.

3 Underlying Data

Choosing the underlying data and retrieving it correctly was a crucial part of our work. Not only we had to (choose) terms that are most used in an LGBT context but also we had to filter the content for relevant articles in our case. There are a lot of terms but not all of them are useful for analysis. In the end we came up with 6 useful terms that delivered enough data to be relevant. These are: *bisexual*, *gay community*, *homosexual*, *lesbian*, *transgender* and *transsexual* making in sum 44,485 articles and a mean of 95.38% of all articles with these search terms in the NYT database.

Why did we choose those words as our data foundation? Firstly they are diverse. They cover a wide variety of alternative gender models without going to much into detail. Secondly they are very prominent. As already said if there aren't enough articles the relevance decreases. Thirdly they cover all timespans researchable via the API, the NYT was founded in 1851. Some of the terms were used very early while others gained importance just in the last time. The number of articles per decade for a given query term is shown in Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7, and Fig. 8.

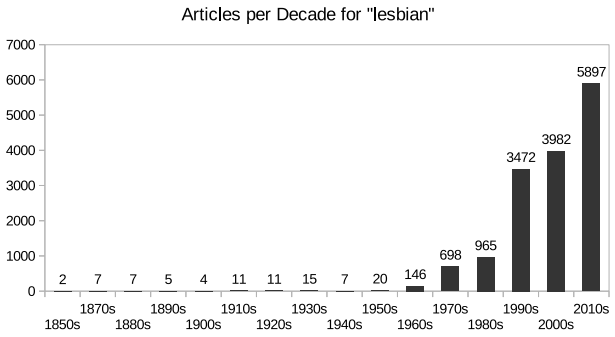

Figure 6: Number of articles for *lesbian*.

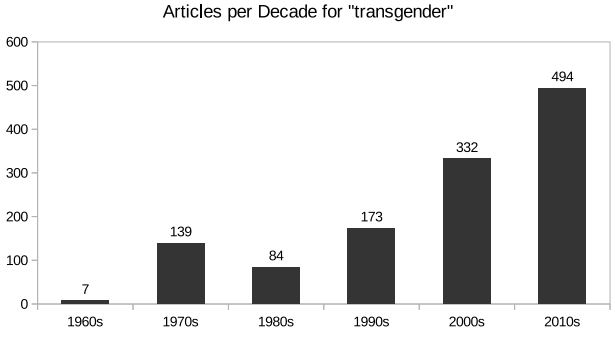

Figure 7: Number of articles for *transgender*.

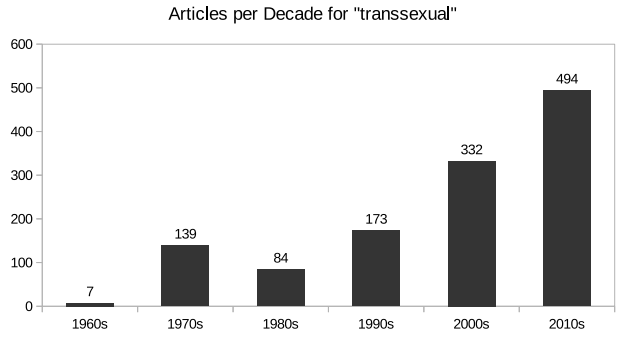

Figure 8: Number of articles for *transsexual*.

Table 1: Size of the data basis.

term	#articles	#paragraphs
bisexual	4,920	1,495
gay	–	10,392
gay community	1,818	377
homosexual	15,086	5,263
lesbian	15,249	5,086
queer	–	162
transgender	6,138	2,973
transsexual	1,229	279

Why didn't we choose other words that are even more prominent for example *gay*? This has several reasons. Let's stay with the word *gay*. It doesn't have an exclusive meaning just in LGBT context. It also means *happy* meaning we would have gotten lots of articles that have nothing to do with our research topic because it was used in the sense of *being happy*. Another reason is, that gay is also a quit common name or part of a name. There are a lot of other terms we didn't use due to reasons like not enough relevance, usage in specific context and or not given researchability.

One thing we thought about analyzing but decided against eventually was a visualization of geographical information. Only 37% of all articles have an assigned geographical location. Thus, we doubted the relevance of such a visualization.

So we have a lot of articles but that's just one part. The other part is sentiment analysis and topic modeling to really get something out of the texts. For sentiment analysis, we focused on the 1980s and 2010s because for both decades have many articles and we wanted to compare two times which we expected to have very different reports on LGBT. An extended sentiment analysis for other decades would have gone beyond the scope of this work.

The analyses are based on the *lead paragraph* attribute of each article. 94% of the articles have such a paragraph. Nevertheless, it is not necessarily the case that a lead paragraph assigned to a certain query term actually contains this query term. For instance, although there are 4,920 'bisexual' articles in the database, only 570 of these articles have a lead paragraph containing *bisexual*. Nevertheless, there are 1495 lead paragraphs containing *bisexual* in the whole database. The difference emerges from the fact that the NYT API does not search within the lead paragraph but only in the headline, byline and body. As our textual analyses are based on the lead paragraph, the count of target terms in all paragraphs in the database is shown in Tab. 1.

Because on target word can occur in more than one lead paragraph, paragraph duplicates are expected. Within the database, 28% of all lead paragraphs occur more than once. Duplicates are eliminated during selection of paragraphs from the database.

Finally, the created corpus has been compared with the Google Books corpus accessed by the Ngram Viewer.[11] Fig. 9 shows the relative frequency for LGBT-related terms in American English books. The upper diagram contains our query terms while the lower one visualizes the occurrence of other LGBT terms. Very prominent is the maximum of *lesbian* (red line) in the late 90s. A closer look at our corpus reveals that there is a similar maximum for *lesbian* but some years earlier. Since newspaper articles are assumed to reflect events more immediately than books, we concluded that it is the same maximum. However, the decrease of *bisexual*, *homosexual*, and *lesbian* since the late 90s, can be only confirmed for *homosexual* in our corpus. The steady increase of *transgender* and *transsexual* is attested in both corpora. The lower diagram shows the frequency of *gay* and *queer*. The characteristic of both terms is that they are present in the literature long before the upper terms occur. Furthermore, as soon as the upper terms emerged, the frequency of *gay* and *queer* decreased while an increase can be observed since the 90s. The hypothesis is that these terms changed their meaning in an infrequent phase from 1940 to 1990 and since that they have been frequently used as LGBT terms.

[11] `https://books.google.com/ngrams`

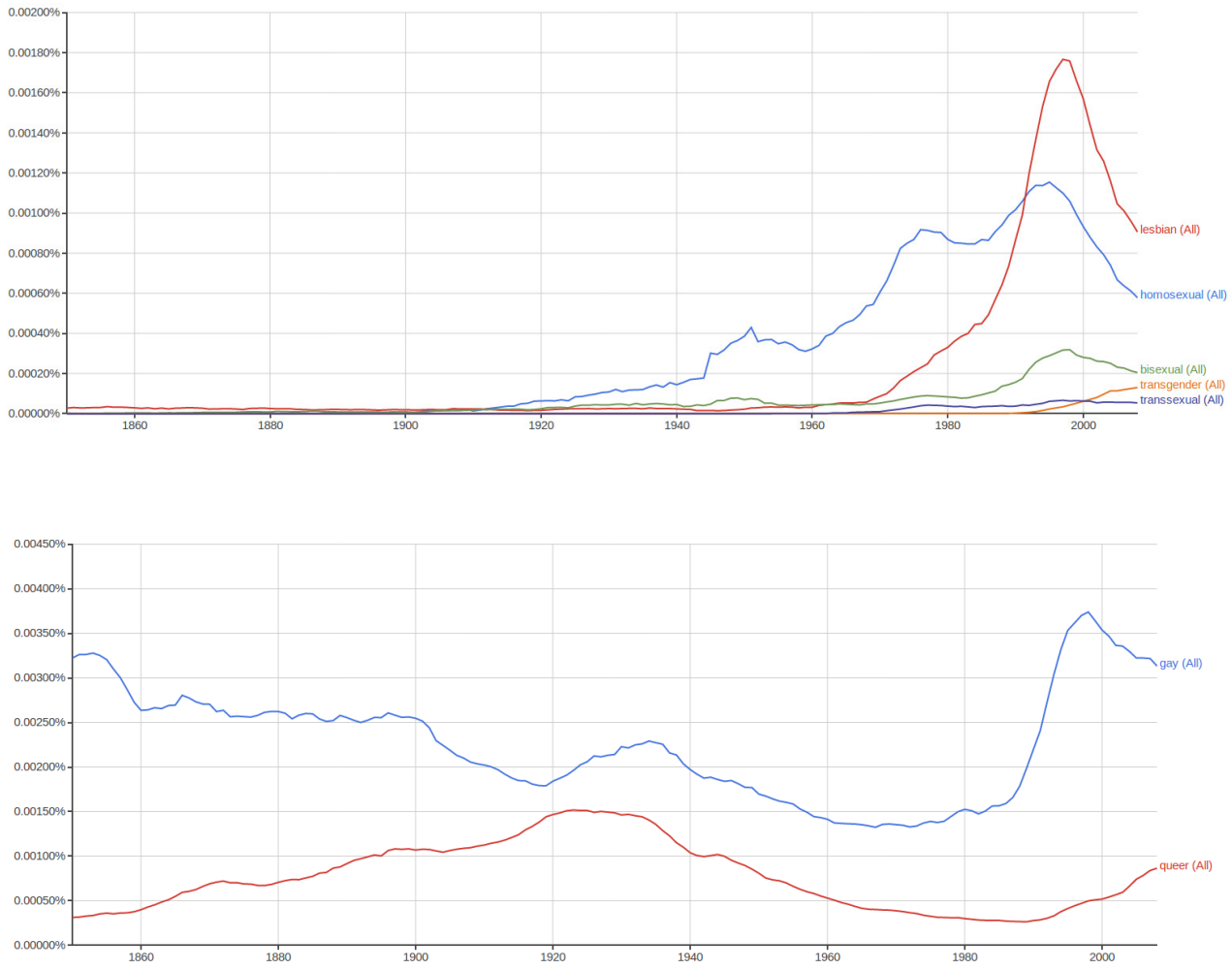

Figure 9: Google Books Ngram Viewer: LGBT-related terms.

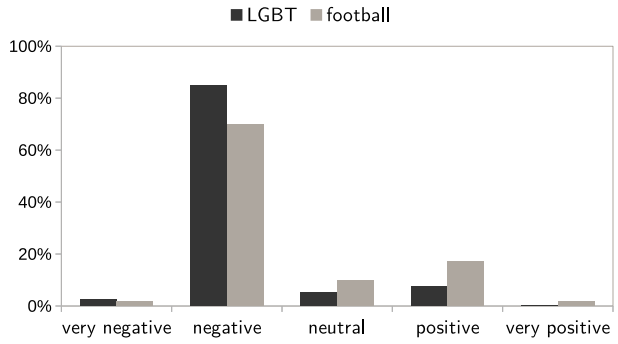

Figure 10: Frequency of sentiments.

4 Results

4.1 Sentiment Analysis

For the sentiment analysis, we compared the sentiments of sentences containing a target word from early newspaper articles (1980s) with ones from current articles (2010s). Furthermore, the sentiment results among the target words were compared. All in all, there were no significant differences within these conditions. On average, 85% of all sentences are assigned the sentiment *negative*. *very positive* is determined to less than 0.1% of all input sentences of a sentiment analysis. The frequencies of the other sentiments are in the range of 2% to 8%. All standard deviations are smaller than 1.5% percent points. There are two explanations for the minimal differences between the conditions. First, the sentiment distribution mainly depends on the genre, i.e., all newspaper articles are written in the same manner. Each article reports the events objectively and with no significant attitude of the author. The mostly negative sentiments are a side-effect of the newspaper ductus. Second, there has been (and still is) a negative reporting for all LGBT news.

To test both explanations, further articles were extracted from the NYT article archive as a baseline for expected positive sentiments. These articles were football news as we assumed that reporting on football is positive. The distributions of sentiments over sentences for LGBT terms and the football baseline are shown in Fig. 10. The *football* data show indeed more positive sentiments than LGBT news. However, about 70% of all sentences[12] have a still a negative sentiment. Thus, we have concluded that negative sentiments are very likely due to the newspaper genre.

It is important to bring to mind that the output of the sentiment analysis is not a sentiment category in the first place. Rather, an vector of probabilities is returned. Hence, it might be possible that the competition between sentiments was very close. For the sentiment vector $v_{s_1} = \langle 0.0, 0.46, 0.1, 0.44, 0.0 \rangle$, for example, the decision for *negative* and against *positive* is very close since the difference in the probabilities is marginal. However, for the vector $v_{s_2} = \langle 0.0, 0.8, 0.1, 0.1, 0.0 \rangle$ it is clear that the sentence s_2 has a negative sentiment.

But also on the underlying probability level, the results are equal for all conditions. The averaged probabilities are summarized in Tab. 2. For the articles from the 1980s (2010s), the standard deviation is less than 1.3 (1.5) percent points. Since the expected probability of each sentiment is $^{100\%}/_{5} = 20\%$ for a random distribution, a probability for *negative* greater than 40% can be seen as a clear result. Nevertheless, there is still a certain probability of about 20% that such sentences are *neutral* instead of *negative*. A similar pattern emerges for *football*.

To summarize, the sentiment analysis showed no differences between target terms or different times. A detailed evaluation of the precision of the sentiments is presented in the discussion.

4.2 Simple Topic Modeling

This is our great topic modeling.

5 Discussion

This is our insightful discussion.

5.1 Implementation Issues

In order to compare the reception of LGBT-related terms in different newspapers, it is necessary to integrate further newspaper articles into the database. Especially for The Guardian[13], the integration into the database is no problem since the Guardian API returns less information than the NYT API, e.g. no publication date or lead paragraph. At the same time, this is a problem because the only information is the web URL to the article. Thus, in order to get the publication date and textual information, the workflow has to be extended by a further request for receiving the HTML page of the full article as well as by a parser

[12] For *football*, the input of the sentiment analysis comprises 125 sentences.

[13] `http://open-platform.theguardian.com/`

Table 2: Average sentiment probabilities.

data	very negative	negative	neutral	positive	very positive
LGBT, 1980s	16.9%	51.0%	20.5%	8.7%	2.8%
LGBT, 2010s	16.5%	51.9%	19.8%	8.8%	3.0%
football	13.6%	44.8%	21.3%	14.4%	5.9%

for extracting relevant information from this HTML document. As soon as such an extension has been implemented, it can be used to get the full article also from the NYT archive. The full article body would enable a more comprehensive topic modeling. Attention should be paid that the Guardian API provides only access to publications dating back to 1999.

A general problem with the newspaper archive APIs is that the access is limited. Assume that there are $150 \cdot 10^3$ articles for a certain query term as it is the case for *gay* in the NYT archive. Since the NYT API allows requesting only $10 \cdot 10^3$ articles per day, collecting the whole result set would take 15 days. The same amount of articles is no problem for the Guardian API where $250 \cdot 10^3$ articles can be requested per day.[14] Additionally, only the first 1200 articles can be addressed within requests to the NYT API. Consequently, in order to get all articles, each call should not return more results than this number. In our workflow, the publication date was restricted in order to obtain an appropriate result set size.

In the present paper, sentences were treated differently. On the one hand, sentences were assumed to be important semantic units. Thus, the sentiment analysis was only executed for the sentence containing the target word. On the other hand, sentence boundaries were ignored for the simple topic model. The underlying argument is that in order to extract the topic in which the target word occurs, it is not only worthwhile to look at the sentence but to consider the context of that sentence as well. Nevertheless, this argument can be also applied to the sentiment analysis. Thus, increasing the window size for sentiment analysis should be evaluated in further investigations. Then, the information on coocuccrences and sentiments can be extended by the distance from the target word. Cooccurrences along a great distance may be less informative as direct neighbors. In addition, the sentiment of the sentence containing the target word has a larger weight then the one of other sentences. Then, an outstanding issue is the aggregation function when combining the results of different distances.

The evaluation of the cooccurrences has revealed that there are still a lot of 'un-meaningful' words, for instance *only*, *some*, *recent* but also *people* and *group*. Such words make it harder to evaluate cooccurrences. Thus, the stop word list should be expanded by *only* and others. Alternatively, a part-of-speech tagger can be used instead of the stop word list. Then, cooccurrences are only considered for particular part-of-speech, e.g. for nouns and verbs. Furthermore, too generic words (*people*) should be excluded from the context words as well.

// TODO: only cooccurrences of sentences? define max distance dynamically? // TODO: count the distance from coWord to the targetWord

5.2 Content Issues

5.3 Future Work

6 Conclusion

This is our awesome conclusion.

References

[Bor08] Stefan Bordag. A comparison of co-occurrence and similarity measures as simulations of context. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 9th International Conference*, pages 52–63. Springer, Berlin, 2008.

[MS03] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing.* MIT Press, Cambridge (Mass.), 2003.

[OD] online Oxford Dictionary. How many words are there in the English language? *https://en.oxforddictionaries.com/explore/how-many-words-are-there-in-the-english-language.* last call: 2017, Feb. 20th.

[Sic74] HS Sichel. On a distribution representing sentence-length in written prose. *Journal of the Royal Statistical Society. Series A (General)*, pages 25–34, 1974.

[14] The Guardian allows $5 \cdot 10^3$ calls per day and a maximum amount of 50 articles per call.

[SPW+13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment tree-bank. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Who did What?

Task	M. Bönisch	T. Haubenreißer	M. Möller
general decisions	✓	✓	✓
programming			✓
basic data analysis			✓
diagram creation		✓	
background recherche	✓	✓	
evaluation of sentiment	✓		✓
topic evaluation	✓	✓	✓
introduction text	✓		
workflow & implementation text			✓
underlying data text		✓	
sentiment results text			✓
homos./bis. topic results text	✓		
lesbian topic results text		✓	
trans*, queer results text			✓
technical discussion text			✓
content discussion text	✓		
future work text			
conclusion text			✓
typesetting			✓