Heidelberg University
Institute of Computer Science
Database Systems Research Group

Final Report

News sentiment development on the example of "Migration"

(Document version for public sharing – personal information redacted)

Team Members: Simon Lüdke
Josephine Grau
Martin Drawitsch

Mentor: John Ziegler

1 Abstract

Although newspaper articles are often regarded as neutral reports, they often expose a non-neutral sentiment when reporting on dividing and emotionally charged topics. In our project we built up a pipeline for sentiment analysis of newspaper articles over time. We concentrated on German news outlets and on the topic "migration". We scraped a large data set of news articles, filtered them for topic relevant articles by a keyword search and partly annotated them. Known sentiment analysis methods were adapted to this German news corpus. We performed three types of sentiment analysis. The first method is a dictionary-based approach based on the SentiWS data. Secondly, we used a BERT model, which was already pretrained and which we finetuned with the self-annotated topic specific data. Lastly, we implemented a Word2Vec model to offer more qualitative analysis by finding association words to our keywords. Our pipeline and interactive visualization allows to compare the different analysis methods and to get insights in the development of sentiment from 2007 until 2019. Our code is available at `https://github.com/text-analytics-20/news-sentiment-development`.

2 Introduction

In democracies such as the federal republic of Germany free press is vital. They inform citizens independently of the state and political parties and thus contribute to the formation of their opinions. It is therefore important to see how they differ in sentiment and if they stand by their beliefs or follow public opinion.
Migration for example is a very dividing topic. Over the years public opinion in Germany ranged from "refugees welcome" to racist stereotypes, demonstrations against the accommodation of refugees and violent attacks on people with a migration background. While the range of opinions is related to the range of the political spectrum, sentiment also changed over time [10].
A comparison of multiple news outlets over time requires the analysis of thousands of news articles. To gain an overview for only one topic by hand would be an incredible amount of work. A well built pipeline on the other hand can easily be adapted to other topics and additional sources.
While similar work was done by [1], they focused on a small time frame and on the national ratio of negative to positive articles. We on the other hand focus on the difference in opinion between news outlets and the change thereof. While the general shift towards a more negative reporting is interesting, it is important to see if news outlets that tend to be more liberal

reacted differently. We also identify the words that are most often associated with refugees and how they change over time.

3 Related Work

Automatic sentiment analysis systems have already been applied to newspaper articles and blog posts in 2007, where the authors of [3] applied the Lydia text analysis system [6] to a collection of articles and computed sentiment scores based on the words found in each article. Each word is looked up in a dictionary and the word's sentiment score ("polarity") is computed by measuring the path between the word and a seed word of known score, where the path length is determined by the number of required "hops" in the dictionary's synonym and antonym database. This system relies purely on word-level sentiments and is not able to identify sentiments that are not immediately clear from the choice of words (this can especially pose difficulties with sarcasm and context-dependent interpretation of words).

More recently, state-of-the-art approaches to sentiment analysis are almost always based on deep learning methods such as BERT [2] (according to most of the top-performing benchmark entries at `https://paperswithcode.com/task/sentiment-analysis`). The currently best performing method on the commonly evaluated SST-2 benchmark (97.5% accuracy) is a very large transformer-based neural network model that was first trained on a large text corpus in an unsupervised fashion and then fine-tuned for solving a multitude of downstream tasks [8]. However, since training, re-purposing and utilization of this method for may require large amounts of compute, we did not regard this particular method as feasible for this project and instead propose a framework that offers variants of a pretrained BERT model [4] and a dictionary-based method inspired by [9]. Although [9] is aimed at analyzing Twitter data, we adapt it to news articles as well because its dictionary-based approach is not per se restricted to short messages.

Previous research has also been published specifically on the topic of sentiment analysis in German-language media on refugees [1]. The authors of this work analyze traditional and social media data collected in a 6-month time frame between 2015 and 2016 and do not clearly differentiate between sources in their results. We instead aim at a much longer time frame of multiple years and will put focus on both the long-term development of sentiment as well as the comparison between different sources. In addition, we do to not limit our analysis to the simple categories "positive", "negative" and "neutral" but present our results in the form of polarity scores that express sentiment as a real number between -1 (negative) and 1 (positive) to enable

a more nuanced view and make it easier to discover trends in reporting (see Figure 3.

4 Methods

During the project we have developed a toolkit for evaluating news sources to understand how the sentiment for a specific topic evolves over time.

The main text analytics task relevant for this project is the sentiment analysis. We chose to evaluate the sentiment by three different methods and rated it with scores between -1 (negative sentiment) and +1 (positive sentiment). A first approach is using a dictionary which assigns to words of an article a sentiment score. The overall score of an article is than calculated as mean value of the word-wise sentiment. As dictionary we used the SentiWS data. We complemented this approach by handling negations to refine the assigned sentiment. Our second approach was to use a pretrained BERT model. We finetuned it with self-annotated training data so that it meets our purpose to differ between sentiments in newspaper articles, which primarily were rated as neutral.

As we dealt here with newspaper articles, we need to consider two main challenges. Firstly, a single article might contain more than one sentiment (e.g. argumentative articles). This is solved rather easily, by taking the mean of the sentiment scores as in the dictionary approach. Secondly, in the special case of our topic "migration", negative sentiment can not indicate the point of view of the article author towards migration, e.g. complaining about missing resources to address the needs of migrants vs. negative stigmatisation of migrants behaviour. Therefore, we complemented the numerical sentiment analysis by the possibility to generate word clouds of associations to keywords by a Word2Vec model.

Our pipeline consists of the following steps:

1. Article scraping
2. Selection of topic relevant articles
3. Dictionary-based sentiment analysis with SentiWS
4. Sentiment analysis with the BERT models
5. Synonyms analysis with the Word2Vec model
6. Timeline visualization

The pipeline proposed for the project is constructed as follows: First, we develop and apply a web scraping system that is used to fetch news article texts and their metadata, given lists of source URLs. The preprocessed article text along with the article metadata is passed through a topic filter. This filter operates by searching for keyword matches in the extracted article keyword metadata and in the text bodies. The topic-filtered article collection is then fed into a sentiment analysis component that yields the corresponding sentiment score that is later visualized. The timeline visualization component then aggregates article sentiment scores and metadata to populate an interactive browser-based dashboard that enables exploratory data analysis.

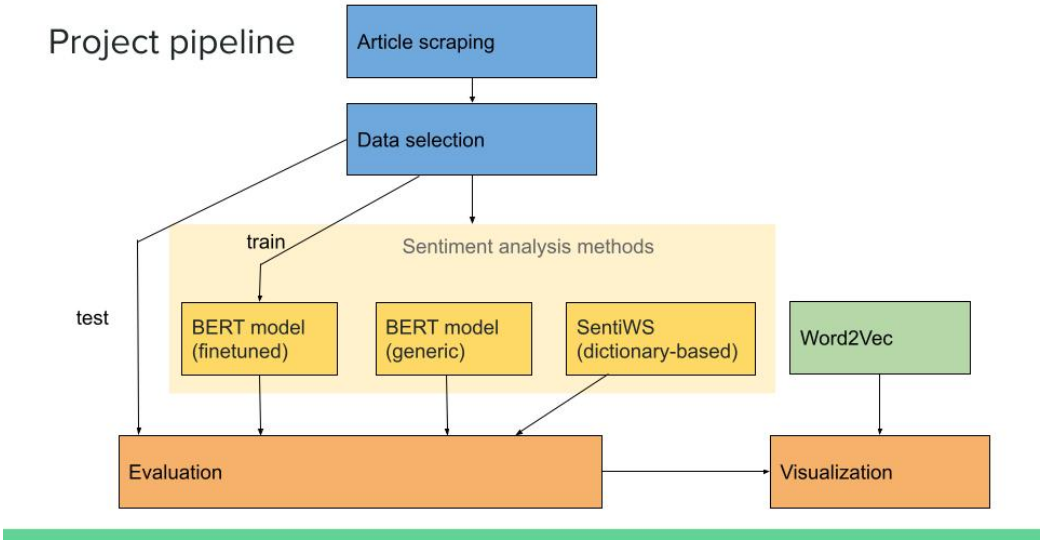


Figure 1: Structure of project pipeline

Figure 1 shows a schematic of the pipeline. Each step will be explained in more detail in the following sections.

4.1 Datasets

4.1.1 Article scraping

Due to the German *Urheberrecht* and *Leistungsschutzrecht*, there is no freely available adequately large dataset of recent German newspaper articles, so we decided to create our own dataset of online newspaper articles, including article texts and metadata. To this end, we have developed a web scraping system that can automatically download articles and metadata given a list of URLs.

Since articles / web pages of even a single news source are not always consistently structured, issues can arise when trying to extract metadata and the article text itself. Metadata may be positioned differently on each page, while article text may be interrupted by more or less related references to other articles ("also read:" or "you might be interested in:") or ads that don't belong to the article text. Also, content of interest could be loaded dynamically via JavaScript and not be extractable by just looking at the source HTML document. We have managed to solve most of these issues by making use of the core functionality of the `newspaper`[1] library. We have still noticed some cases where the scraping routine did not return the expected article text, but this was not a general issue due to our subsequent filtering/selection steps that discards such incomplete articles.

Our initial approach to collecting news data for our analysis was based on directly crawling public news websites for links to articles, collecting their URLs and then downloading, parsing and categorizing each article based on the collected URLs. We implemented these steps with help of the `newspaper` Python library and made use of the JSON serialization format for storing each scraped article with its text and metadata (publication date, title, keywords found in HTML meta-tags etc.) in a machine-readable format for the later processing steps.

With this code we have successfully scraped hundreds of articles from news websites including `https://spiegel.de`, `https://focus.de`, `https://de.rt.com`, `https://bild.de` and `https://welt.de` and have stored the resulting article collections in separate files, which unfortunately cannot share here due to copyright reasons.

However, we found that despite our initial expectations, the newspaper library's site crawling functionality only can find a very limited number of articles on each site and only finds recently published articles. Although we could use this limited collection for testing and building the later steps of our analysis pipeline, it is of course not sufficient for our intended large-scale historical analysis.

Since we did not find a suitable method to automatically crawl all relevant news sources for article URLs and associated publication dates, we have therefore decided to use the publicly available news datasets from `https://wortschatz.uni-leipzig.de/en/download/german`. The main content of these datasets consists of single, scrambled sentences without their articles' contexts or metadata, but the included `deu_news_*_-sources.txt` files do contain URLs and publication dates in most cases, starting from year 2007 (earlier years' datasets only contain the domain information). Since the

[1] `https://github.com/codelucas/newspaper`

included raw text data was not in usable form, we used the included lists of sources and dates to collect our corpus of articles, which consists of articles published between 2007 and 2019 (except 2016, which was missing from all dataset variants) [2] Since for each year there are at least 1 million articles from a diverse set of German-language sources in the dataset, we expected to be able to build a sufficiently large collection of articles for our project. Due to limited storage capacity and internet speed we chose to use the *100k* versions of these datasets instead of the full *1M* ones, so in total we had a list of 12,000,000 source URLs that we used as inputs for our scraping system.

At first we encountered the problem that scraping such a large number of articles was prohibitively slow (it would have taken one to two weeks of continuous downloading), but by distributing the scraping tasks across 32 parallel processes in our code, we managed to reduce the total time needed to scrape all data to approximately 14 hours.

After discarding unreachable, deleted, paywalled, very short and otherwise unsuitable articles during this routine, we arrived at a total of 769,331 general news articles.

4.1.2 Selection of relevant articles

A newspaper article was selected to contribute to the sentiment analysis pipeline according to certain keywords. We decided to perform a case-insensitive search for words including the following substrings, as they have a rather neutral sentiment and define the topic clearly (examples for matching words are given in parentheses):

- "flüchtling" (→ "Flüchtling", "Flüchtlingspolitik", "Flüchtlingskrise", "geflüchtet", "Geflüchtete(r)", ...)
- "migra" (→ "Migrant(in/en)", "Migration", ...)
- "asyl" (→ "Asyl", "Asylpolitik", "Asylverfahren", ...)
- "einwander" (→ "Einwanderung", "einwandern", ...)
- "geflüchtete" (→ "Geflüchtete")

Some newspaper sources already expose the metadata attribute "keywords", which in this case is used for this search. Otherwise at least one of the keywords must appear in either the title and the text or the article description (if this attribute is available). Both attributes "text" and "title"

[2] For the list of files that we used, refer to the Appendix of this document

are available for all considered news sources. As German newspapers focus on the migration towards Europe, articles of related, for us non-relevant content, are the exception (e.g. migration from Mexico to the United States) and negligible.

After filtering our article collection through this selection process, we now had approx. 22,000 relevant articles as input for the next steps of our analysis pipeline.

4.1.3 Labeled news sentiment dataset

For evaluation of our methods and for training of our neural network model (see subsection 4.3 we have created our own sentiment analysis dataset based on search keyword-containing text sections[3] of the relevant articles that we had collected earlier as described above. Our data used for training and validation consists of hand-annotated sentiment data (classification into neutral, positive, negative). To prevent data leakage, we performed the annotations on a small reserved subset of our relevant articles that was excluded from the final evaluations. We randomly split this subset into three equally sized parts, one for each team member who then annotated their part with help of the annotation tool that we created for this purpose[4]. After our individual annotations were finished, we merged and shuffled them randomly and finally created a training-validation split with a validation data ratio of 10%. Thus we used 1485 sections for finetuning of the pretrained BERT model and reserved 165 snippets for validation and comparison of the different methods. An overview of the label distributions is shown in Table 1. The slight class imbalance towards relatively few positive samples is to be expected due to the nature of the topic of interest.

Label	# Samples
Positive	267 (18.0%)
Negative	615 (41.4%)
Neutral	603 (40.6%)

(a) Training set

Label	# Samples
Positive	27 (16.4%)
Negative	65 (39.4%)
Neutral	73 (44.2%)

(b) Validation set

Table 1: Label distribution in our news sentiment dataset.

[3] A section consists of a sentence that contains one of the keywords, and in addition the next sentence, to provide more context. If the last sentence also contains a keyword, the subsequent sentence is also included, and so on

[4] `https://github.com/text-analytics-20/news-sentiment-development/blob/main/annotation/sentiment_annotation.py`

4.2 Sentiment analysis with SentiWS and negation handling

A straightforward way to determine the sentiment of a sentence or a text is to look at the sentiment each individual word holds using a dictionary holding fixed sentiment values for as many words as possible. The main advantage of this method over machine learning is that it does not require prior training to be applied to a specific topic. This makes it much more flexible and easier to use. It also means that it can be used without a labeled data set. As our dictionary we use the `spacy-ws` extension to `spacy`, which is based on the SentiWS corpus of the Leipzig University. This corpus contains around 1,650 positive and 1,800 negative adjectives, adverbs, verbs and nouns as well as their inflections. The positive and negative polarity of the words is given in the range of [-1;1].

The topic for sentiment analysis is defined by the list of keywords (see subsubsection 4.1.2. To make sure that the sentiment is connected to our topic, we only consider sentences that contain at least one word that is partly identical with any of the keywords. The sentiment value of a sentence is then the sum of all weighted words in the sentence excluding the keyword. The words we search for are excluded to avoid the bias that a keyword with positive or negative sentiment would induce. An example for this is "Flüchtling", which is weighed negatively in itself. We then calculate the sentiment of the article as the sum of all sentiments of these sentences and normalize it by dividing by the number of words with sentiment value.

To refine this dictionary approach, a syntactical analysis can be added to detect negated words or negated sentences. We used the linguistic features which are offered by `spacy`, in particular the part-of-speech tagging[5]. It detects not only the part of speech of a word, but also its function in the sentence (e.g. auxiliary or main verb) and the dependence to other words. Therefore, it offers a basis for detecting negations. We defined a number of German negation words: "nicht", "kein", "nirgends", "nirgendwo", "niemand", "niemals", "nirgendwohin", "nie"[6] and included all their flexions ("keine", "keiner", "keines", ...) by stemming. A single word is declared to be negated if one of these negation words is syntactically dependent on it. A whole phrase is negated if a negation word is dependent on the the main verb of this phrase. In these cases the sentiment score of a word changes its sign, i.e. it is multiplied by -1.

[5]More information about spacy's linguistic features: `https://spacy.io/usage/linguistic-features`

[6]see e.g. `https://www.worddive.com/grammar/de/deutsche-grammatik/9-verneinungswoerter/`

The detection of negated phrases was tested on 30 typical German sentences, including 7 negated sentences[7]. The function detected 29 out of the 30 correctly. The one which was not detected correctly is in fact a sentence consisting of two main clauses: “Dir kann man nichts schenken, du hast ja schon alles.” A detection depends therefore also on the definition whether phrases like this are negated or not or on a detection of more complex syntactic structures. As this example shows, the approach is of course far from being exact. If the structure of a sentences is too complicated, the sentiment score might be detected faulty. In addition, the sentiment score might not be perfectly symmetric, so that a word does not have the same absolute value sentiment score as its negation. Furthermore, cases where the same word appears more than once in the same phrase, negated and non negated, are not covered. However, the handling of negations clearly refines the assignment of a sentiment and should be taken in consideration when applying a dictionary approach.

4.3 Sentiment analysis with BERT models

4.3.1 Generic pretrained model for sentiment labeling

The "BERT (generic)" method in our pipeline is directly based on the pretrained BERT model that was published in [4]. This neural network model uses the architecture of [2] and retrains it for the downstream task of sentiment analysis of German texts from several different sources, including hotel, movie and app reviews as well as Wikipedia and news articles. Since we want to perform analysis on news articles, it is important to note that all news articles are labelled as neutral in the datasets used for training [4] because the authors appear to assume a neutral sentiment in news articles. This assumption is of course problematic for our application because it wrongly biases the model towards classifying all news articles as more neutral than their actual sentiment.

4.3.2 Finetuning the BERT model for news data

In order to alleviate this bias, we created an additional version of the model by training it with the additional training data that we describe in **??** (TODO: Create section describing dataset). The values of notable training hyperparameters that we found to yield the best results on the validation set can be found in Table 2; for all other hyperparameters we use the default training

[7] `https://deutschlernerblog.de/100-saetze-reichen-fuer-ein-ganzes-leben/`

setup of the `transformers` library (version 4.3.2) [8]. For more details on the training process, please refer to our Colab notebook `bert_finetune.ipynb`[9]. As this kind of transfer learning with limited data and low learning rate is commonly called "finetuning", we name the resulting model "BERT (finetuned)".

We selected the best-performing model snapshot w.r.t. validation loss, which was reached after completing the 3rd training epoch (further training lead to slight overfitting – compare the top left graph and bottom right graphs of Figure 2).

Hyperparameter	Value
Optimizer	AdamW [5]
Minimum LR	0
Maximum LR	$8 \cdot 10^{-6}$
LR warm up steps	120
LR decay steps	630
Epochs	7
Batch size	20
Weight decay	0.01

Table 2: Training hyperparameters

To make this finetuned model easily available for use in our project pipeline and facilitate future work, we have published its weights, code and usage instructions at `https://huggingface.co/mdraw/german-news-sentiment-bert`.

4.3.3 Sentiment polarity estimation

Both the original model and our finetuned version were trained with a classification head that produces a softmax distribution over the three possible sentiment label values "positive", "negative" and "neutral". This means that the model output consists either of an estimated probability distribution of these discrete labels or, if applying an additional argmax operator after it, the output is the label ID with the highest probability estimate. Instead of either of these outputs, we are actually interested in an estimate of the sentiment *polarity*, a real number between -1 and 1, so we have modified

[8] See default values at `https://huggingface.co/transformers/main_classes/trainer.html#transformers.TrainingArguments`

[9] `https://github.com/text-analytics-20/news-sentiment-development/blob/main/training/bert_finetune.ipynb`

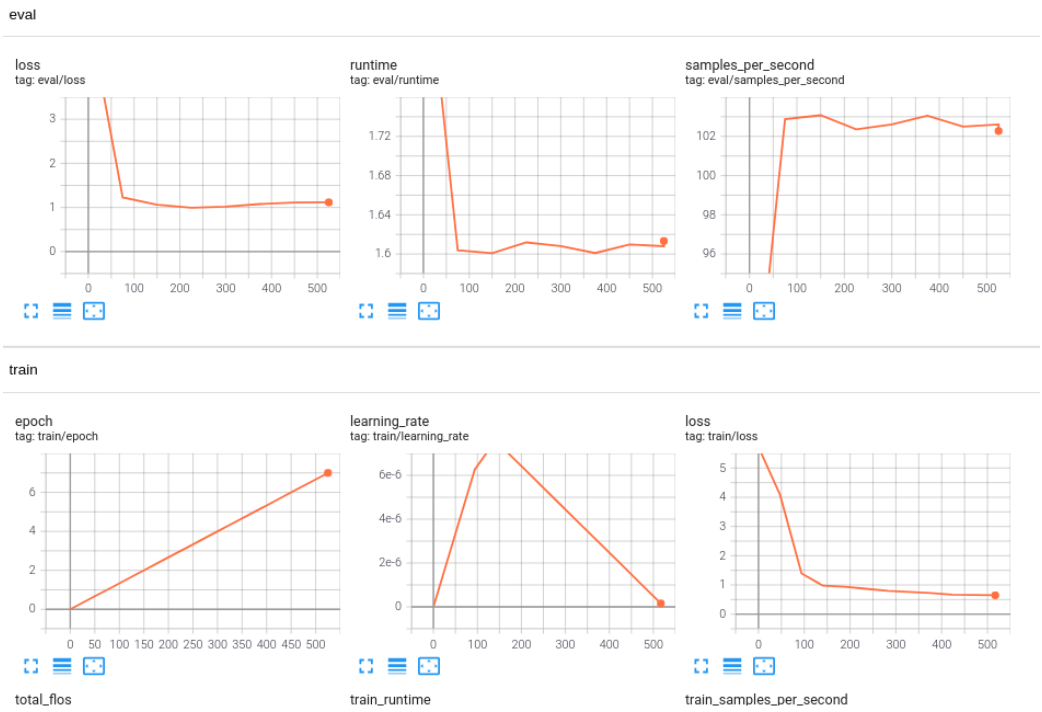


Figure 2: Tensorboard scalar dashboard after the training with our data has finished. The top left and bottom right curves show the validation and training losses calculated at the end of each epoch (i.e. every 75 training steps). The learning rate warm up/decay scheme can be seen in the bottom center curve.

the source code of the model to calculate such a score from the probabilty estimates by subtracting the "negative" probability (n) from the "positive" probability (p) (polarity = p - n) [10]. We ignore the "neutral" output of the model as it is implicitly encoded as (1 - p - n). The correspondence between discrete labels and polarities can be seen in Table 3.

4.4 Word2Vec model

While the above mentioned methods give quantifiable results it is hard to verify the accuracy without a huge labeled data set. Another possible approach is to use Word2Vec to create a list of the most similar words in a text and compare how they change with time or publisher. Word2Vec [7] works by learning a vector embedding of a word as a by considering the words that

[10] See `https://github.com/text-analytics-20/news-sentiment-development/blob/main/sentiment_analysis/bert.py#L50`

Sentiment polarity	Sentiment label
-1.0	1 (negative)
0.0	2 (neutral)
1.0	0 (positive)

Table 3: Sentiment representation by a polarity or corresponding discrete label. Each row shows equivalent representations.

appear next to it. The similarity of two words is then calculated as the cosine similarity between the two vectors. This means that the list generated by using Word2Vec contains words that are usually used with the same surrounding words as the keyword we study. While those do not necessarily have a close connection to the keyword, they at least give an idea of what is used in a similar context as the keyword and therefore of the context itself. To study the differences between years and publishers we created subsets of our data that fulfills a certain condition (year of publication, publisher or both) and then visualized them as word clouds.

4.5 Visualization

Our project offers two types of visualizations. Firstly, the Word2Vec model described above returns a certain number of synonyms or associations words which are summarized in a word cloud. It takes into consideration the similarity score, i.e. displays words larger which have a higher similarity to the word which is currently examined.

Our second type of visualization is an interactive dash board, showing the sentiment scores of the articles over the time. It offers the possibility to examine and compare the sentiment development of different publishers and for the three methods used for the sentiment analysis: the BERT models (generic and finetuned) and the dictionary approach with SentiWS data. By hovering over the data points, the titles of the newspaper article are displayed, which makes it easier to verify and compare the used methods. Furthermore a trend line is shown, giving an indication of the development of sentiment over time.

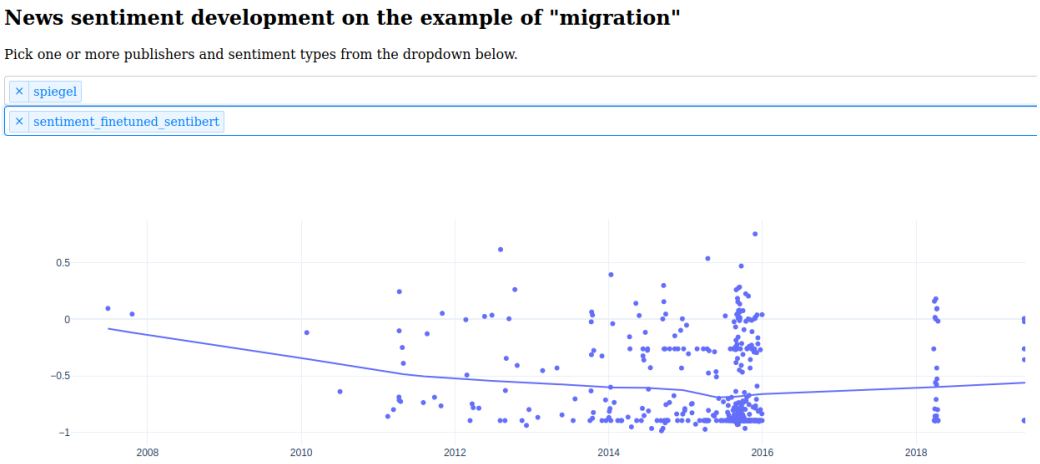


Figure 3: Timeline visualization

5 Results

5.1 Dictionary based Sentiment Analysis

We will now take a look at the results of the dictionary based sentiment analysis. For this a few selected sentences are shown together with their sentiment scores. Those sentiment scores are not normalized since we normalize at the article level. The sentiment value that is shown therefore is the sum of all word sentiments.
The dictionary approach can yield good results since many positive sentences use positive words like “gerettet” (saved), “jemanden versorgt” (cared for somebody), “Erfolg” (success):

Der 85-Jährige erzählt gerne - und häufig - jedem, der es hören möchte, dass er die Einwanderer früher eigenhändig gerettet und versorgt hat. **(0.349)**

Vom syrischen Azubi zum fertigen Kaufmann für Büromanagement Mohammed Masmas (28) und Eyad Isper (32) kamen 2015 als Geflüchtete aus Syrien – und legten eine Erfolgsgeschichte hin. **(0.3757)**

Sentences with negative sentiment on the other hand often use negative words. The reason for this is that they either describe something that is perceived as bad or want to make something seem bad. An example for this is “schlechter” (worse) in the following sentence.

In Quartieren mit einem hohen Anteil von SchülerInnen mit Migrationsh-

intergrund seien die Bildungsabschlüsse schlechter und die Ausbildungsquote gering. **(-0.3562)**

However, we also found texts that were misclassified. False negatives seem to be common because flight is often associated with tragic fates. Many articles (such as the example below) that are classified as negative in fact criticize and describe the terrible conditions in the home country, on the way or mistreatment of refugees in the country of arrival. The sentiment in the following sentence is negative but directed against the treatment of refugees by the italien government not the refugees themselves.

In den vergangenen Wochen geriet Italien erneut international ins Kreuzfeuer der Kritik mit direkten Abschiebungen von Bootsflüchtlingen nach Libyen, die auf See aufgegriffen worden waren. **(-0.5308)**

5.2 Word2Vec similarity analysis

We calculated the most similar words using the Word2Vec approach described in subsection 4.4. As a very small data set could influence the sentiment we chose to concentrate on well known newpapers of which we have a lot of articles (Spiegel, Focus, Taz). When we now take a look at the word clouds created for these three publishers (see Figure 4), it is indeed interesting how different the collection of words are. First of all there are a few words that can be expected to be used synonymously with "Flüchtling". E.g. "Syrer" (Focus,Spiegel) and "Eritrea" (Focus) being the nationality or home country of a huge proportion of refugees, "Balkan" (Focus) describing a region that is a common route for migrants, and "ankommen" (Focus) or "Ankunft" (Taz) which meen to arrive. Interestingly these seem to be most common in the word cloud for "Focus".
But there are also a lot of words with negative sentiment. Especially the paper "Spiegel" seems to often use words such as "gefährlich" (dangerous), "Vorfall" (incident) and "Verbrechen" (crime) in a similar context as "Flüchtling". While this in itself is not enough to judge the sentiment this newspaper wants to project it is noteworthy that the known left-wing newspaper "Taz" mainly uses words that imply innocence and weakness such as "minderjährig" (minor), "schwanger" (pregnant), "Zuflucht" (refuge) or "Kirchenasyl" (church asylum).
As was already discussed in subsection 5.1 negative words can be used in a positive context (for example to describe the misery of others and why they should be granted asylum) and positive words can be used in a negative

context. Still the difference between the newspapers is striking and implies a positive sentiment of the Taz and a negative sentiment of the Spiegel.

(a) Focus

(b) Spiegel

(c) Taz

Figure 4: The 10 most similar words for "Flüchtling" calculated using articles of three important German newspapers from 2007 to 2019.

We repeated this step for different years to investigate a possible shift in sentiment (see Figure 5). Since we do not have any data for 2016 we selected the years 2013, 2014 and 2015 as there is a large amount of data for those years and immigration started to become a prominent topic. There seems to be a reduction of positive associations such as "behandeln" (treat), "verwandt" (related), "schwanger" (pregnant) or "minderjährige" (minors) with time. 2014 there are much less of them then 2013 and "abschiebung" (deportation) is added. This is not in the 10 most similar words for 2015 but the associations seem more bureaucratic and less about the vulnerability.

(a) 2013

(b) 2014

(c) 2015

Figure 5: The 10 most similar words for "Flüchtling" calculated using articles for each of the years 2013, 2014 and 2015.

5.3 Sentiment analysis

5.3.1 Quantitative Comparison

Since our project deals with a rather subjective matter, automated evaluation of the results is not straight-forward. For quantitative evaluation we needed to hand-annotate a number of text examples to compare our sentiment analysis methods with. For the evaluation of our different sentiment analysis approaches we used 165 text samples from our hand-annotated validation data. We compared the sentiment scores of each method with the annotation.

Our methods all predict sentiment polarities (floating point numbers between -1 ("negative") and +1 ("positive")) whereas the annotated data expresses sentiment in terms of discrete labels (0: positive, 1: negative, 2: neutral), so they cannot be compared directly without transforming either the prediction or the target label into the representation of the other one (following the mapping seen in Table 3).

For our quantitative evaluation we used two different metrics that both aim to measure how much the predicted sentiment deviates from the hand-annotated target sentiment.

Firstly, we evaluate the mean absolute error (MAE) of the predicted polarities. Given a sequence of N target polarities p_i (from manual human

annotation) and corresponding polarity predictions $\hat{p}_i$ (from one of our automated sentiment analysis methods), the MAE metric is defined as

$$MAE = \frac{1}{N} \sum |p_i - \hat{p}_i|$$

This metric measures deviations in a fine-grained fashion and differentiates between minor and major errors.

As an alternative metric we also propose the mean category error (MCE), which is is defined given a sequence of N (discrete) target class labels y_i (from manual human annotation) and corresponding class label predictions $\hat{y}_i$ (from one of our automated sentiment analysis methods):

$$MCE = \frac{1}{N} \sum e(y_i, \hat{y}_i)$$

with

$$e(y, \hat{y}) = \begin{cases} 0 & y = \hat{y} \\ 1 & y \neq \hat{y} \end{cases}$$

We obtain the class label predictions $\hat{y}_i$ by looking up the associated class from polarities $\hat{p}_i$ in a set of intervals that define to which polarity each sentiment class corresponds to:

$$\hat{y}_i = \begin{cases} \text{“}negative\text{”} & \hat{p}_i < -\frac{1}{3} \\ \text{“}neutral\text{”} & |\hat{p}_i| \leq \frac{1}{3} \\ \text{“}positive\text{”} & \hat{p}_i > \frac{1}{3} \end{cases}$$

	Dictionary approach	BERT generic	BERT finetuned
MAE (polarity)	0.57	0.56	**0.52**
MCE (label)	0.56	0.56	**0.47**

Table 4: Mean error values of the different methods on our validation set. *MAE* is the mean absolute error between predicted polarities and target polarities (real numbers). *MCE* is the mean category error between predicted labels and target labels (discrete).

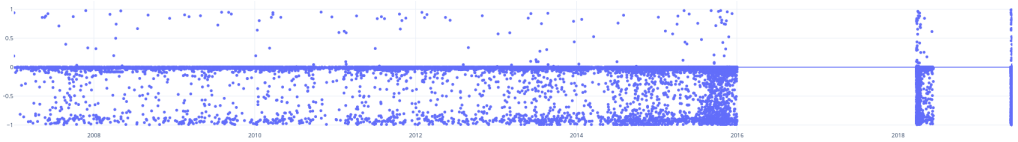
In contrast to the MAE, the MCE metric only differentiates between correct and incorrect classifications without regarding the exact values.

In terms of both metrics, we observe that our finetuned BERT model achieves the best results in total, whereas the dictionary approach and the pretrained generic BERT model deliver less accurate results (see Table 4).

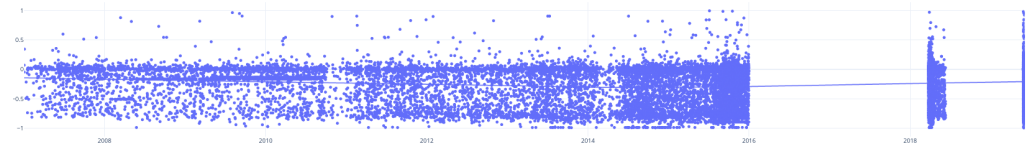
5.3.2 Qualitative Comparison

We created timelines for different newspapers for all three sentiment models as was described in subsection 4.5. For the dictionary based model and the generic BERT model, the trend lines nearly always stayed close to neutral (see Figure 6) . Trends are only clearly visible using the fine tuned BERT model. We will therefore only use this model for the discussion of trends. This is feasible because it was also shown to be the most accurate when used with our labeled data set (see subsubsection 5.3.1). For the following analysis we concentrated on the years 2014 and 2015 because this time frame was the peak of the "refugee crisis" and during this time there are sufficient and well distributed data points for all three newspapers.

(a) Dictionary approach using senti-ws.



(b) Generic Bert Model



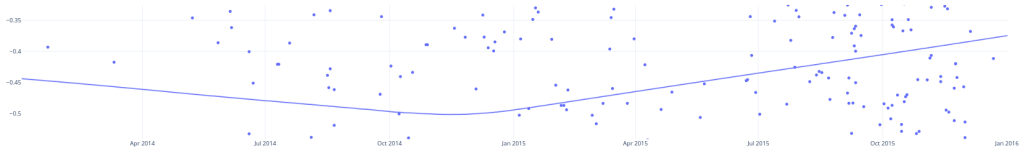
(c) Finetuned Bert Model

Figure 6: The figure shows all data points that we have. From 2007 until 2019, evaluated by the three different sentiment models.

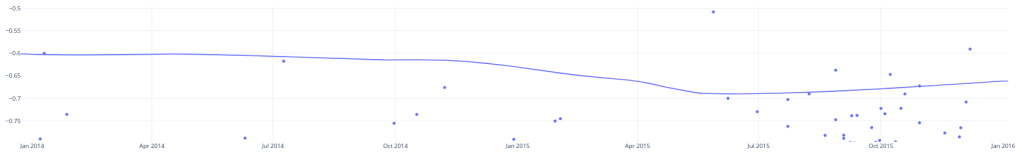
We again look at the three newspapers we already discussed in subsection 5.2. The Taz, which stood out for its positive association words is constant at a comparably high sentiment value of approximately -0.24. This slightly negative value does not mean that the newspaper is anti refugees because it is likely that there is a bias towards negative sentiment. Because of this we will not discuss the absolute values but only the differences between different newspapers. The Spiegel for example, which had many negative association words, also has the most negative sentiment values (around -0.65). The sentiment value is constant for the most part of 2014 but decreases by

0.1 between November 2014 an Mai 2015 only to slowly increase again. This makes it seem as if there was some kind of event in November 2014 that changed public opinion but decreased in importance with time. The problem with this conclusion is that Focus, which again is situated between the other two (around -0.45), shows the opposite trend. Here the sentiment value decreases linearly till November 2014 and then linearly increases. The only event with which this could correlate is the start of the migration-critical Monday demonstrations of the Pegida movement in October 2014.

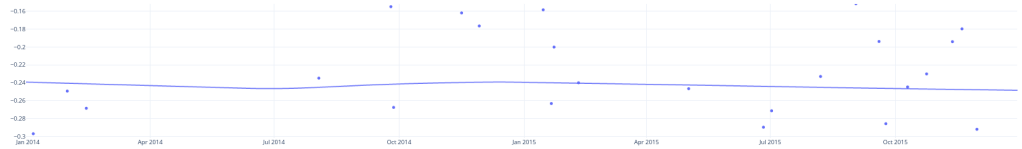
Our data is insufficient to assess if there is indeed a causal effect and how the reactions of the two newspapers are the interpreted. The first impression would be that the Spiegel got influenced by the right wing protests while the Focus had an adverse reaction. But this is only speculation since a decrease in sentiment value could also result form negative articles about the protests and vice versa.



(a) Focus



(b) Spiegel



(c) Taz

Figure 7: The trend lines for the results of the fine tuned BERT model for three different prominent German newspapers. The time line shows the years 2014 and 2015. Only a small range around the trend line is shown to make the trend more clear therefore not all data points that influence the trend line are visible and the scale is different for each diagram.

6 Conclusion and future work

In our project we explored different sentiment analysis approaches to analyse the sentiment towards "migration" in German newspaper articles. These methods offer first insights in the data and are valuable tools to analyse the sentiment. However, to determine cause and effects and to state the sentiment differences more precisely, future work would be necessary.

Determining the sentiment of newspaper articles is in general harder compared to other text corpora like customer reviews or tweets. The text is much longer and can include different, opposing sentiments, e.g. in an argumentative article. Furthermore, newspaper articles tend to be written in a more formal and neutral style in order to inform objectively with facts and less opinion.

However, the main difficulty is to evaluate negative sentiment. Especially in the topic "migration" negative sentiment can mean both, hostility towards refugees, but also stating grievances. To differ more precisely, the BERT model could be trained with data tagged additionally with a hostility label.

An expansion of our project could be a generalization of the pipeline to other topics. In this case we would only need to adapt the filtering of articles by different keywords and the training of the BERT model with topic specific annotated data. Due to the recent crises, the topic "covid" would be an interesting choice.

Furthermore it would be interesting to examine the causes of sentiment changes. How much do key events influence the development of sentiment (in our case the for example the sexual assaults at the New Year's Eve 2015-16)? Or is there a more general development of sentiment independently of topic, e.g. a tendency towards negative sentiment over time, when the topic is not new any more.

A further step could be a more precise analysis of the behaviour of media in comparison with public opinion. Is the opinion influenced by the newspaper articles or do they only reflect public opinion? Is the sentiment of public opinion in general more positive as newspaper articles try to gain attention by exaggeration or selection? These questions would need far more research. Approaches could include a sentiment analysis of reader comments, comparison to sentiment development of social media content and including opinion research surveys.

Anti-plagiarism confirmation

We hereby declare that all material in this report and in our project is our own work except where there is clear acknowledgement or reference to the work of others.

Note on workload distribution

We have split the work for the project into different focus areas and assigned them to the team members as follows:

- Web scraping and news data collection: Martin
- Pretrained BERT model adaptation: Martin
- BERT model training setup and finetuning: Martin
- Quantitative evaluation of sentiment analysis methods: Martin
- Requirements documentation: Martin, Simon
- Code for manual sentiment data annotation: Josephine, Martin
- Interactive sentiment data visualization dashboard: Josephine
- Word cloud visualizations: Josephine
- Filtering relevant articles: Josephine
- Organization of documentation and group meetings: Josephine
- Designing and implementing the dictionary-based sentiment analysis approach: Simon, Josephine
- Integration of components into a pipeline: Simon
- Pipeline configuration system: Simon
- Testing: Simon

In total, all team members have contributed equally to this project.

References

[1] Gerhard Backfried and Gayane Shalunts. Sentiment analysis of media in german on the refugee crisis in europe. In *International Conference on Information Systems for Crisis Response and Management in Mediterranean Countries*, pages 234–241. Springer, 2016.

[2] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[3] N. Godbole, Manjunath Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs (system demonstration). In *ICWSM*, 2007.

[4] Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. Training a broad-coverage German sentiment classification model for dialog systems. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1627–1632, Marseille, France, May 2020. European Language Resources Association.

[5] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[6] A. Mehler, Y. Bao, X. Li, Y. Wang, and S. Skiena. Spatial analysis of news sources. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):765–772, 2006.

[7] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[9] Karsten Tymann, Matthias Lutz, Patrick Palsbröker, and Carsten Gips. Gervader -a german adaptation of the vader sentiment analysis tool for social media texts. 09 2019.

[10] Bastian Vollmer and Serhat Karakayali. The volatility of the discourse on refugees in germany. *Journal of immigrant & refugee studies*, 16(1-2):118–139, 2018.

Appendix: News sources

The article source files (`-sources.txt` files) containing URLs for our article collection (see subsubsection 4.1.1) can be obtained from `https://wortschatz.uni-leipzig.de/en/download/German)`. They are located inside of the `.tar.gz` files listed there. For this project we used the following archives:

- `deu_news_2007_100k.tar.gz`
- `deu_news_2008_100k.tar.gz`
- `deu_news_2009_100k.tar.gz`
- `deu_news_2010_100k.tar.gz`
- `deu_news_2011_100k.tar.gz`
- `deu_news_2012_100k.tar.gz`
- `deu_news_2013_100k.tar.gz`
- `deu_news_2014_100k.tar.gz`
- `deu_news_2015_100k.tar.gz`
- `deu_newscrawl_2017_100k.tar.gz`
- `deu_newscrawl_2018_100k.tar.gz`
- `deu_newscrawl-public_2019_100k.tar.gz`