Comparing the Performance of Lexicon-Based Sentiment Tools across Domains

Friederike Schreiber
Universität Potsdam
Matriculation number: 810847
schreiber6@uni-potsdam.de

Anna-Janina Goecke
Universität Potsdam
Matriculation number: 777707
goecke@uni-potsdam.de

Abstract

For this project multiple sentiment tools were implemented in R to analyze the suitability of domain-specific lexicons to handle input data from other domains. Accordingly, we compared the performance of the sentiment dictionaries on various corpora coming from different domains. We cannot report statistically significant differences between the performance of the lexicons. However, there is a high performance range of the tools over the different data sets, indicating that the type and quality of the data is a discriminating factor when performing sentiment analysis. We propose to use high quality data sets to obtain satisfactory performance rates. Also, we found VADER to be the most promising choice for straightforward sentiment analysis on mixed domains.

1 Introduction

Sentiment Analysis is probably one of the best-known Text Mining applications to extract subjective information in input texts. With its help, input data can be automatically classified into positive, negative and neutral sentiment. It is commonly used tool to analyze product reviews or social media data.

This project aims to explore the performance of lexicon-based sentiment tools on various domains (political and financial texts, Twitter data and product reviews). We implemented three different sentiment dictionaries (Afinn, VADER, and Lexicoder Sentiment Dictionary) in a black box approach and tested their performance on five different corpus data sets. The sentiment tools were chosen due to their prominence in the context of lexicon-based sentiment approaches and the fact that each of the dictionaries is attuned to a specific domain. Each sentiment tool was implemented as a black box to make the project replicable and extendable for upcoming tasks including further data sets or lexicons. Hence, no additional syntactic rules or dictionary entries were considered for the implementation. In the evaluation of the tools, we report that no significant performance differences of the tools on the same data set were obtained. The lexicons performed very similar when being applied on one data set. We could not observe any statistically significant difference in their performance (ANOVA p-value > 0.9). In contrast, the performance of a tool on various data sets showed significant differences (ANOVA p-value < 0.001). The following sections will address previous work on lexicon-based sentiment analysis, its evaluation, as well as the comparison of multiple tools. We will explain how we collected and preprocessed the data sets and give a brief overview over the implementation steps. Finally, we will discuss our findings and propose potential modifications for future work. Our implementation, including additional data set configurations, results and plots can be found on our GitHub page[1].

2 Related work

Most of the previous work comparing different lexicon-based sentiment approaches concentrated on the comparison of the performance across the tools rather than evaluating its behavior within distinct domains. Musto et al. (2014) for instance investigate the quality of four lexical sentiment resources on two Twitter data sets. They propose a fine-grained approach which splits each tweet into micro-phrases and combines the polarity of those micro-phrases to a final sentiment score. This was tested with an implementation of four different configurations including normalized and emphasized[2] versions. Their results clearly show a difference

[1] https://github.com/ajgoecke/sentiment_tools

[2] They added weights to sentiment words being associated with certain part-of-specch categories.

in the performance of the sentiment tools within each corpus, indicating that there is indeed a discrepancy when analyzing sentiment of the same data with different tools. The same was found by Khoo and Johnkhan (2018) who compared several sentiment dictionaries by examining the polarity of Amazon product reviews. Khoo and Johnkhan (2018) also included an experimental set up which assigned certain weights to specific part-of-speech categories. Accuracy rates of the tools ranged from 0.615 to 0.77 and indicated a considerable difference in the performance of the tools on the reviews data. More work on this has been made by Al-Shabi (2020) who compares five sentiment dictionaries, including Afinn and VADER, on two Twitter data sets. He obtains the best accuracy with the VADER lexicon (72% for Stanford data and 65% for Sandars data) , while Afinn is reaching accuracy scores of 65% (Stanford data) and 62% (Sandars data).[3] For all lexical resources being documented in Al-Shabi (2020), accuracy rates across the data sets ranged from 0.53 to 0.72, suggesting an inequality in the performance of sentiment tools even when applied on data sets belonging to identical domains. A study by Gonçalves et al. (2013) not only focuses on the comparison of various lexicon-based methods with respect to the prediction of sentiment strength or polarity but also examines the interrelatedness of a lexicon's accuracy and the coverage[4] of the lexicon itself on a given data set. They computed the coverage of all sentiment methods across the data sets and found that the percentage of data most lexical resources could identified is very low. Additionally, they compared agreement of the different tools concluding that the lexicon's predicted polarity ranges widely, indicated by agreement scores spreading from 33% to 80% and therefore are even likely to output contrasting sentiment for the same input text in some cases. Likewise, Miazga and Hachaj (2019) compare three sentiment dictionaries with respect to their coverage on various data to determine usability indicators of the sentiment lexicons and Gatti et al. (2015) inspect the trade-off between the lexicon's performance and its coverage. Gatti et al. (2015) claim manually annotated lexical resources to produce high precision while lacking for the coverage aspect and propose a new resource of about 155.000 features with both high precision and high coverage by including inter alia the removal of stop words in their preprocessing step.

3 Sentiment Tools

Typically, lexicon-based approaches of sentiment analysis make use of predefined sentiment lexicons, i.e. word lists of positive and negative words. Those lists are either binary and comprise a word's polarity (positive or negative) or contain an associated sentiment score or so-called sentiment strength (Nielsen, 2011). Sentiment lexicons consisting of continuous sentiment scores can be used to calculate an overall sentiment strength of an input text by looking up terms in the lexicon and aggregating the scores of all sentiment words within a text to obtain a final sentiment score (Kannan et al., 2016). In contrast, binary lexicons can be applied to an input text to retrieve its ”bare” polarity, e.g. by counting the appearances of positive and negative sentiment words. Lexicon-based analyses commonly depend on adjectives and adverbs to capture a text's Semantic Orientation (SO) (Gupta and Agrawal, 2020). Techniques exploiting lexicon-based approaches to attain the polarity of a text are generally very simple by means of implementation and replicability (Sadia et al., 2018).

In the following, we introduce the three sentiment tools that will be used in this project to perform sentiment analysis on our data.

3.1 Lexicoder Sentiment Dictionary

The Lexicoder Sentiment Dictionary (LSD) (Young and Soroka, 2012) is a publicly available sentiment dictionary specifically adapted to political content. The lexicon itself consists of terms from Roget's Thesaurus, the General Inquirer (GI), and the Regressive Imagery Dictionary (RID) and is composed of two major lists, one with positive words and another one with negative words, with a total of 4.567 lexical features. Additionally, the LSD consists of two supplementary lists containing negated positive and negated negative terms[5].

By reason of its binary fashion the LSD is especially suitable for lexicon-based sentiment approaches trying to determine the polarity or SO of a text rather than its sentiment strength.

[3] We only report the results of VADER and Afinn here, since those two lexicons are of major interest within the scope of this project.

[4] Coverage here searches to assess the input texts whose sentiment has been classified.

[5] Both lists together sum up to additional 4.581 entries. This is the approach of Young and Soroka (2012) to handle negation.

Corpus Name	Type of Content	Label Type	Rating Scale
Product Reviews	Reviews from Amazons industrial and technology section	star rating by customers	1 (bad) to 5(good)
Book Reviews	Reviews from childrens books on Goodreads	star rating by customers	1(bad) to 5 (good)
ParlVote	UK parliamentary speeches	voting behavior	binary accept/ reject
Twitter Corpus	Tweets during a republican debate in 2016	machine annotated	positive/ neutral/ negative
Financial News	Newspaper headlines about one company	human annotated	positive/ neutral/ negative

Table 1: Overview of the corpora.

3.2 VADER

VADER (Hutto and Gilbert, 2014) is a rule-based sentiment tool attuned to microblog-like domains, i.e. social media data and reviews texts. It contains lexical entries from the Linguistic Inquiry and Word Count (LIWC), Affective Norms for English Words (ANEW), the GI and is enriched by social media abbreviations, acronyms, slang words and facial expressions. The VADER tool exploits five simple heuristics to compute sentiment strength of an input text: i) punctuation, e.g. exclamation marks as intensifier of the SO, ii) capitalization, to increase the strength of sentiment terms, iii) degree modifiers, i.e. degree adverbs may emphasize a word's individual scoring, iv) conjunctions as polarity shifters, e.g. ”but”, and v) negation, i.e. the polarity of the three items following a negation is being inverted. All 7.500 lexical features contained within the VADER dictionary are rated on a scale from -4 (extremely negative) to +4 (extremely positive).

3.3 Afinn

Afinn (Nielsen, 2011) is a sentiment word list comprised of 2.477 lexical entries, manually labeled by Finn Årup Nielsen with a valence score between -5 and +5. The dictionary is tailored to identify sentiments of social media contexts and was initially created from a set of negative and positive words and then enlarged with words from the Original Balanced Affective Word List by Greg Siegle, slang terms and acronyms from the Urban Dictionary, and word lists by Steven J. DeRose. The sentiment strength is obtained by summing the sentiment scores of sentiment-laden words in a text and dividing it by the number of words contained in the according input.

4 Data

To evaluate our lexicons, we choose five pre-built corpora. Details can be found in table 1. The corpora were chosen to cover a range of different domains with two product review corpora, one social media Twitter corpus, one corpus including financial headlines and one corpus containing political speeches. It is to note that the Twitter corpus consists of tweets about a political debate. To keep a black box approach, we only applied light preprocessing which consisted of removing non-English characters and discarding incomplete examples. We also excluded non-English entries. Since the corpora showed an unequal rating distribution and had considerably different sizes, we choose to work with a subset of each corpus consisting of 1000 randomly sampled examples with an equal rating distribution. For the Twitter and finance corpus which featured positive, neutral, and negative labels two subsets were created: one containing only positive and negative labels (binary set) and one additionally containing neutral labels (ternary set). For the Twitter corpus, the binary set holds only 600 entries due to limitations in the corpus size.

5 Implementation

For the implementation of this project, we used RStudio Team (2020). Since the idea was to make this project reproducible for future work or potential additions of sentiment tools and domains, we implemented the sentiment tools as they were pro-

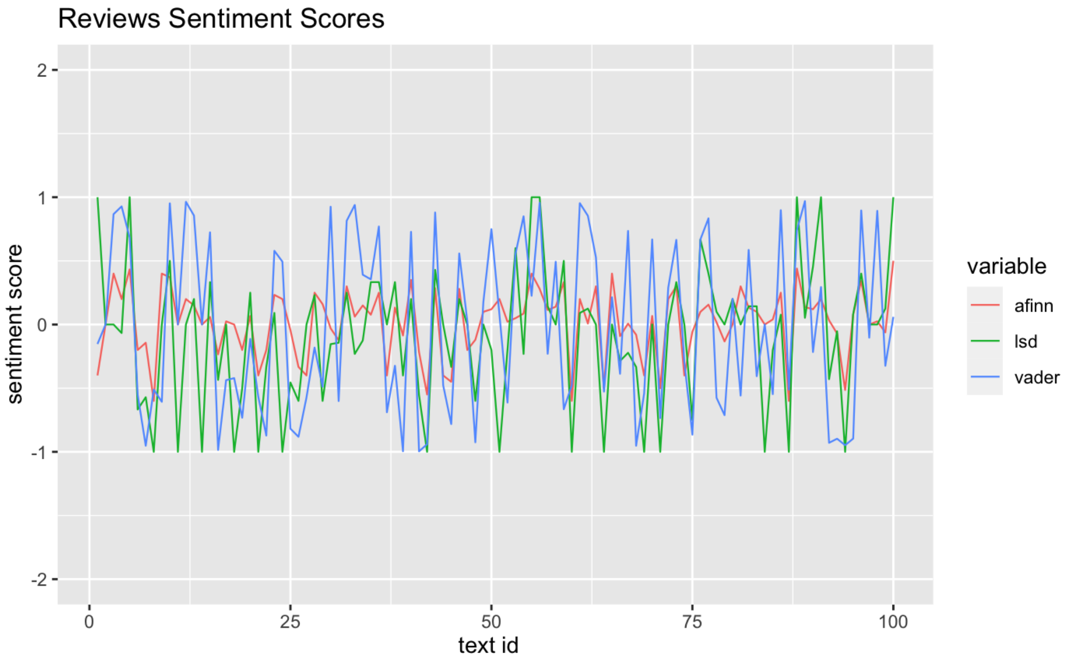

Figure 1: Normalized sentiment scores for the first 100 instances of the book reviews corpus.

vided by the according libraries. Respectively, no changes to the original tool or additional rules to handle negations or intensifiers were made. The code, including its documentation, can be found on GitHub. We provide R-Notebook files which make it easier to retrace the implementation structure[6].

5.1 Sentiment Analysis

As already mentioned, the sentiment tools are implemented straightforward through the according libraries. Since the data sets have already been cleaned in the preprocessing step, they can be fed directly into our `get_sentiment` function. For each sentiment dictionary the sentiment strength (Afinn, VADER) or polarity scores (LSD) are being tracked and saved as a new column in the output data frame. Scores are rounded to three digits from the decimal point. Missing values (*NA*) are transformed to zeros given that we consider non-recognized texts to be of *neutral* polarity and to ease further processing and conversion of the sentiment scores. See figure 1 for an example of the different sentiment scores each dictionary produces[7].

5.2 Normalization

To be able to compare the outputs of the different sentiment dictionaries on an equal scale, normalization plays a major role when dealing with the sentiment scores. The `vader` library in R Core Team (2017) provides the output in form of a `compound score` which is calculated by retrieving the sum of the sentiment scores of each word in the lexicon. The scores are first being modified by the inherent rules of the tool and then outputted in a normalized scale between -1 (extremely negative) and +1 (extremely positive)[8]. According to Hutto and Gilbert (2014) the `compound score` is the most used metric for sentiment analysis research, given that the positive, negative and neutral proportions of a text should add up to 1.
Contrastingly, the `quanteda.sentiment` package which we used to implement the LSD includes the `sent_relpropdiff` (i.e. ”relative proportional difference”) function which uses the following formula to compute the difference between positive and negative occurrences (Lowe et al., 2011):

$$\frac{pos - neg}{pos + neg} \tag{1}$$

Equation 1 scales the polarity given by the LSD to a continuous rating system from -1 to +1.

[6] `https://github.com/ajgoecke/sentiment_tools`; see `README` for further information.

[7] Plots for all data sets can be found on GitHub via `implementation/plots/`

[8] More information about the compound score and the `vader` library can be found here: VADER GitHub Page.

Whereas VADER and LSD are automatically normalized within the application of the tools via the libraries, Afinn scores are not normalized by definition. Afinn is, similarly to LSD, implemented via the `quanteda.sentiment` package. The library includes the option to average over sentiment-laden words only by using the `dictionary` function. To be able to then compare all tools, we used a customized minimum-maximum normalization technique to range final sentiment scores between -1 and +1.

Min-max normalization is a commonly used method to normalize values to a specific range (Han et al., 2012). To apply for the fact that Afinn scores vary between -5 and +5, we considered these two values as the basis for the new minimum (-1) and the new maximum (+1). The original formula is given in equation 2 (Han et al., 2012), where the normalization maps a value to the range $[a, b]$:

$$x = \frac{x - min(x)}{max(x) - min(x)}(b(x) - a(x)) + a(x) \quad (2)$$

This equation is transformed to the following to apply for our case of -5 and +5 being the old minimum and maximum values of the Afinn dictionary and -1 and +1 being the new minimum and maximum:

$$x_{afinn} = 2 * \frac{x+5}{5+5} - 1 \quad (3)$$

We evaluated equation 2 and 3 on our data sets to see which one would give us with the best performance of the Afinn lexicon. Both normalization methods produce equivalent sentiment scores (in terms of accuracy and f1 scores)[9]. Since equation 3 ensures that words or texts not being contained in Afinn are labeled with *neutral*, i.e. a score of 0, we decided on using the customized version given in equation 3 for our evaluation of the lexicon performance.

5.3 Coverage

For the upcoming evaluation, we additionally wanted to compute the coverage of each sentiment dictionary on the present data (Gonçalves et al., 2013; Miazga and Hachaj, 2019; Gatti et al., 2015). We define coverage as the ratio of texts or tokens (with respect to the corpus) being identified and scored by a lexicon. If a lexicon tool does not recognize any word within an input text, this text is considered to be ”not covered” by the according sentiment dictionary. The same strategy applies for single tokens: within our `get_sentiment` function we included the option to examine the sentiment scores with regard to each token. Whenever a token could not be identified by the lexicon, i.e. it is not contained within the lexicon, it is considered to be not covered. During the valuation of our way to compute the coverage for each token, we realized that the coverage ratio may be more meaningful when removing stop words beforehand. Consequently, we can report coverage ratios per text, token, and token with prior stop word removal. We report an overview of the coverage scores in table 2.[10]

Corpus	Afinn	LSD	VADER
Books	90.70	87.70	96.50
Twitter	49.10	49.90	57.10
ParlVote	82.70	83.60	93.10
Amazon	77.00	78.20	87.60
Finance	50.50	45.00	67.00

Table 2: Coverage ratio per text in percentage for each corpus.

6 Evaluation

6.1 Binary and Neutral Test Conditions

As a first evaluation step we calculated the accuracy and F1 score of the lexicons to get an impression of overall performance on the different domains. Since the ParlVote corpus is based on binary labels we decided to split the sentiment results in two evaluation conditions. Testing once with only binary labels (positive and negative) and once with additional neutral labels included. For the neutral labels we decided on a threshold by selecting the one that gave us the highest average accuracy and F1 score in the range from +-0.01 to +- 0.25 around 0, going in steps of 0.01[11]. The highest accuracy was reached for a threshold of 0.08, while the highest F1 score was reached for 0.2. We decided on using 0.08 as a threshold, since this value was close to

[9] Accuracy of equation 2 is 0.614, F1 score is 0.633 and accuracy of equation 3 is 0.613, F1 score is 0.629. The difference is not significant.

[10] Additional tables containing the coverage ratios for token-wise computation (with and without stop word removal) can be found on GitHub: `implementation/results/coverage/`

[11] I.e. sentiment scores betwenn -0.08 and +0.08 are labeled *neutral*, while values below -0.08 are *negative* and values higher than +0.08 are *positive*.

Corpus	Afinn	LSD	VADER	Mean
Amazon	**0.653** / 0.700	**0.662** / 0.704	**0.691** / 0.740	**0.669** / 0.715
Books	**0.660** / 0.752	**0.688** / 0.756	**0.704** / 0.780	**0.684** / 0.763
ParlVote	**0.488** / 0.526	**0.489** / 0.549	**0.494** / 0.583	**0.490** / 0.553
Twitter	**0.548** / 0.490	**0.550** / 0.485	**0.552** / 0.526	**0.550** / 0.500
Finance	**0.699** / 0.642	**0.681** / 0.577	**0.656** / 0.674	**0.679** / 0.631
Average	**0.610** / 0.622	**0.614** / 0.614	**0.619** / 0.661	**0.614** / 0.632

Table 3: Accuracy (first value in bold) and F1 score (second value) for best configuration for data with binary labels.

Corpus	Afinn	LSD	VADER	Mean
Amazon	**0.547** / 0.405	**0.552** / 0.409	**0.583** / 0.431	**0.561** / 0.415
Books	**0.616** / 0.427	**0.618** / 0.418	**0.622** / 0.404	**0.612** / 0.417
Twitter	**0.362** / 0.289	**0.381** / 0.302	**0.370** / 0.297	**0.371** / 0.296
Finance	**0.492** / 0.395	**0.454** / 0.355	**0.494** / 0.395	**0.480** / 0.382
Average	**0.504** / 0.379	**0.501** / 0.371	**0.517** / 0.382	**0.508** / 0.377

Table 4: Accuracy (first value in bold) and F1 score (second value) for best configuration for data with neutral label.

the threshold value suggested to use by Hutto and Gilbert (2014)[12]. The gold labels in the corpora were divided as follows to match the test condition: For the Twitter and finance corpora we used either the binary or ternary subset. For the Amazon and books reviews corpus, which both featured ratings on a 1 to 5 stars scale, we determined the rating division by best performance. For the binary condition, a rating of 1 or 2 stars was considered negative and a rating of 3 stars or above was considered positive. We looked into 20 randomly selected entries of 2 and 3 stars rated examples which backed up this cutoff point, but it is to note that this cutoff only describes a trend and sentiment varied strongly over the ratings. For the neutral condition a 3 star rating of the Amazon corpus and a 2 star rating of the book reviews corpus was counted as neutral. The ParlVote corpus was excluded from the neutral test condition since it only includes binary labels.

6.2 Performance Results

The results of the binary and neutral test condition appeared to be slightly different with the binary condition reaching a better performance (average accuracy of 0.614 and F1 score of 0.632) over the neutral condition (average accuracy of 0.508 and F1 score of 0.377) but sacrificing labeling specificity. The detailed results can be found in the tables 3-4. Overall, the performance between the lexicons is very similar with only minor performance differences being not statistically significant. ANOVA analysis showed a p-value of 0.986 for the binary condition and of 0.976 for the neutral test condition. In contrast, the difference in performance between the corpora was quite notable. ANOVA analysis showed a p-value of < 0.001 for both test conditions. Tables 5 and 6 show the detailed results of a Tukey test on the corpus pairs.

6.3 Corpus Configurations

Since the overall performance was lower than expected, we decided to implement additional corpus configurations which included further preprocessing steps. The conditions ended up being the following:

1. Base condition
2. Lemmatization and stop word removal
3. Lemmatization
4. Stop word removal

To remove stop words, we used the stop words list provided by the `tm` library. From this list we removed any negations to not interfere with the negation handling of LSD and VADER. Additionally, we added a list of the top neutral words[13] across the lexicons to the stop word list. Lemmatization was done with the `lemmatization` function

[12] The `vader` library uses a threshold of 0.05.

[13] These lists were obtained for the upcoming qualitative analysis by computing the Top-N words for each polarity label with respect to each sentiment tool. The lists can be found on GitHub via: `implementation/results/word_counts/`

Compared Pair	p-Value
Books-Amazon	0.786
ParlVote-Amazon	< 0.001
Twitter-Amazon	< 0.001
Finance-Amazon	0.942
ParlVote-Books	< 0.001
Twitter-Books	< 0.001
Finance-Books	0.994
Twitter-ParlVote	0.009
Finance-ParlVote	< 0.001
Finance-Twitter	< 0.001

Table 5: Correlation comparison with binary labels.

Compared Pair	p-Value
Books-Amazon	0.008
Twitter-Amazon	< 0.001
Finance-Amazon	0.001
Twitter-Books	< 0.001
Finance-Books	< 0.001
Finance-Twitter	< 0.001

Table 6: Correlation comparison with neutral labels.

of the `textstem` library. The results of these test conditions only showed minor, not statistically significant changes. With the base condition being the best configuration for the binary condition and condition 4 being the best configuration for the neutral condition.[14]

6.4 Similarity Comparison

In addition to the performance comparison, we also computed the cosine similarity of the examples of the lexicons and created a list of the most divisive examples for each corpus by computing the absolute distance[15]. The cosine similarity indicated moderate similarity between the lexicons which is in line with previous results showing no significant performance differences between the sentiment rating of the lexicons.

Through qualitative analysis we observed some discrepancies in the lexicon rating, even though their effect does not seem strong enough to impact the overall performance significantly. Hence, we explored the instances that produced the most dissimilar ratings. We noticed that one factor contributing strongly to this arose from the fact that Afinn does not generate appropriate scores when encountering negations. The tool itself does not include negation handling. Nielsen (2011) proposes to manually add heuristics to take care of negated elements, however this is not included in our black box implementation. Thus, especially shorter entries which featured very few sentiment terms led to a strong disagreement of sentiment scores between the tools. An example for this is *"So am I the only one not impressed?" Not great not what I expected* which was correctly identified as being negative by LSD and VADER, while Afinn incorrectly labeled this entry as positive as a result of missing negation management. Apart from this the data shows expected patterns: the lexicons struggled with entries that did not contain any sentiment words that were included in the dictionary and therefore could not be rated. Furthermore, instances that are incomplete or only consisted of very few sentiment-laden terms are frequently given an incorrect polarity label.

As already mentioned, we also created lists of the most frequent words for each corpus and each polarity category, i.e. positive, negative, and neutral, to qualitatively explore our data sets and potentially find an explanation for the performance issues.[16] Anyway, we cannot report any unexpected findings.

7 Discussion

We could not find significant performance differences between the lexicons. Our implementation could not replicate the findings of Musto et al. (2014) who found lexicons to perform diversely or Khoo and Johnkhan (2018) who reported that the performance of lexicon tools differed when being applied to the same data set. It is to mention that they investigated a different set of sentiment lexicons as the ones we explored within this project. Additionally, they adjusted the lexicon tools by adding weights. This could indicate that customizing the lexicon tools could lead to higher accuracy rates and could improve the performance of the dictionaries. Similar to Al-Shabi (2020),

[14] The results of the different test conditions can be found on GitHub via: `implementation/results/results_test_configurations/`

[15] Please see `implementation/results/similarity comparison tables/`

[16] The plots of the most frequent words with respect to each polarity can be found on GitHub via: `implementation/plots/`

we found VADER to perform best in nearly any configuration and on any data set. That is why we propose VADER as the best choice for a general purpose sentiment analysis, even though it is originally thought to be applied on social media content. Furthermore, the VADER tool was very straightforward when it comes to the implementation part, given that the library provided us with an elaborated rule-based system with inherent negation handling and normalization steps.

As indicated above, the results show significant difference in the performance of a lexicon on the different corpora. Surprisingly, we could not observe the LSD (which is specifically attuned to political texts) to perform better on the corpora containing political content (Twitter and ParlVote) compared to VADER or Afinn. Furthermore, Afinn and VADER did not perform significantly better on the Twitter corpus albeit both tools are tuned to work best with social media data. This leads us to the conclusion that even though the lexicons are meant to be attuned to specific domains, they do not certainly give better results on the domain they are adapted to. In other words, in contrast to what we expected, the lexicons did not reveal an advantage for the specific domain for which they were designed. Since we examined only a few corpora we cannot claim with certainty that our results are indicative of a larger trend of general purpose lexicons being as good as domain-specific lexicons in their domains. Especially the poor performance of the sentiment tool on some of the corpora may have contributed to the lexicons showing a similar performance.

The results imply that the performance on the Twitter and the ParlVote corpus was particularly poor in comparison to the other corpora. Hence, we examined possible explanations for this. After taking a closer look at the Twitter corpus, we noticed that the removal of any information other than the pure text led to problems with the tweet comprehensibility. Sentiment of Twitter data is not only covered through text but also through included gifs, videos, and memes. Also, the polarity of a tweet is highly embedded in the context of users reacting to other users. Therefore, obtaining sentiment only through the text is difficult even for humans. The poor coverage of only 50 percent may also reflect the limitation of the lexicons to pick up sentiment information in the tweets. We believe that the poor text quality is one of the main contributors to the mediocre performance of the Twitter corpus. To give some examples of text from the Twitter corpus where context is clearly missing due to removed embedded input: *"If there's one thing to take away from last night's"*, *"RT This picture taken during"*. The poor performance of the Twitter corpus has to be taken into account when interpreting the performance results. Hence, further evaluation on other social media corpora may be advisable.

For the ParlVote corpus we had a closer look on the process that was used to obtain sentiment. The sentiment of this corpus is not human annotated but coupled to the voting behavior of the members of the British parliament. Thus, since the voting behavior does not necessarily capture the text's polarity, we suspected that the lexicons could have problems with predicting accurate sentiment. Furthermore, a correlation analysis between the party membership of the speaker and the party bringing the motion showed a correlation coefficient of 0.64 towards similar voting behavior. In contrast, the correlation of the voting behavior and the predicted results of the sentiment tools showed a value of -0.02. The strong correlation with the party membership indicates that an example rating cannot completely be grasped through the sentiment of the text. Additionally, with the sentiment being restricted to only binary labeling, misclassifications for ratings containing no sentiment-laden term are inevitable. To give an example, the sentence *"They have been sent away"* got a positive gold label, while *"What does the hon Gentleman say?"* was labeled to be negative. In both cases the lexicons could not pick up any sentiment-laden words.

In contrast to the Twitter corpus, the ParlVote corpus shows a high coverage with an average of 86.47% which implies that the performance problems of the lexicons do not stem from poor text quality or low lexicon coverage but reflect that voting behavior may not work as an adequate indicator for polarity.

8 Conclusion

We investigated the use of pre-built sentiment tools over several corpus domains. Our results suggest that domain-specific sentiment dictionaries do not have an edge over general purpose lexicons when the tools are implemented as a black box.

Since our project dealt with only a very small number of corpora, we cannot certainly tell whether the performance difference between the corpora is domain dependent or reliant to the specific data sets that we used. We assume the high accuracy rates of the lexicons on the Amazon product reviews and the book reviews data to be domain-related since review texts in most cases closely reflect the user's sentiment. In this case, there is no need of additional context to predict the polarity.

After all, it seems to be crucial to use a high-quality sentiment corpus for the exploration of lexicon-based tools. Otherwise, the usage of inappropriate corpus data could lead to very poor performance of the sentiment tools. We therefore suggest using human-annotated data for the evaluation of sentiment dictionaries. For future work related to this project it may be interesting to expand the number of corpora and to include multiple corpora for each domain to see if the observed effect continues. Besides, we assume the fact that we treated the sentiment tools as a black box to be questionable and to be a potential explanation for the poor performance. Even though this was done on purpose to design a replicable study, tweaking the dictionaries by adding specific sentiment terms or including further heuristics such as negation handling and modifier words may be a promising avenue to explore.

References

G. Abercrombie and R. Batista-Navarro. 2020. Parlvote: A corpus for sentiment analysis of political debates. *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5073–5078.

M. A. Al-Shabi. 2020. Evaluating the performance of the most important lexicons used to sentiment analysis and opinions mining. *IJCSNS*, 20(1):1.

K. Benoit, K. Watanabe, H. Wang, P. Nulty, A. Obeng, S. Müller, and A. Matsuo. 2018. quanteda: An r package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30):774.

L. Gatti, M. Guerini, and M. Turchi. 2015. Sentiwords: Deriving a high precision and high coverage lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, 7:1–1.

P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha. 2013. Comparing and combining sentiment analysis methods. In *Comparing and combining sentiment analysis methods*, volume Proceedings of the first ACM conference on Online social networks, pages 27–38.

N. Gupta and R. Agrawal. 2020. Chapter 1 - application and techniques of opinion mining. In S. Bhattacharyya, V. Snášel, D. Gupta, and A. Khanna, editors, *Hybrid Computational Intelligence*, Hybrid Computational Intelligence for Pattern Analysis and Understanding, pages 1–23. Academic Press.

J. Han, M. Kamber, and J. Pei. 2012. 3 - data preprocessing. In J. Han, M. Kamber, and J. Pei, editors, *Data Mining (Third Edition)*, third edition edition, The Morgan Kaufmann Series in Data Management Systems, pages 83–124. Morgan Kaufmann, Boston.

R. He and J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

C. Hutto and E. Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

S. Kannan, S. Karuppusamy, A. Nedunchezhian, P. Venkateshan, P. Wang, N. Bojja, and A. Kejariwal. 2016. Chapter 3 - big data analytics for social media. In R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, editors, *Big Data*, pages 63–94. Morgan Kaufmann.

C. Khoo and S. B. Johnkhan. 2018. Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. *Journal of Information Science*, 44(4):491–511.

W. Lowe, K. Benoit, S. Mikhaylov, and M. Laver. 2011. Scaling policy preferences from coded political texts. *Legislative Studies Quarterly*, 36(1):123–155.

P. Malo, A. Sinha, P. J. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

J. McAuley, C. Targett, Q. Shi, and A. van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA. Association for Computing Machinery.

J. Miazga and T. Hachaj. 2019. Evaluation of most popular sentiment lexicons coverage on various datasets. In *Proceedings of the 2019 2nd International Conference on Sensors, Signal and Image Processing*, SSIP 2019, page 86–90, New York, NY, USA. Association for Computing Machinery.

C. Musto, G. Semeraro, and M. Polignano. 2014. A comparison of lexicon-based approaches for sentiment analysis of microblog posts. In *A Comparison of Lexicon-based Approaches for Sentiment Analysis of Microblog Posts.*, volume DART@ AI* IA, pages 59–68. Citeseer.

F. Å. Nielsen. 2011. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903.

R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

RStudio Team. 2020. *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA.

A. Sadia, F. Khan, and F. Bashir. 2018. An overview of lexicon-based approach for sentiment analysis. In *An overview of lexicon-based approach for sentiment analysis*, volume 2018 3rd International Electrical Engineering Conference (IEEC 2018), pages 1–6.

M. Wan and J. J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM.

M. Wan, R. Misra, N. Nakashole, and J. J. McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2605–2610. Association for Computational Linguistics.

L. Young and S. Soroka. 2012. Affective news: The automated coding of sentiment in political texts. *Political Communication*, 29(2):205–231.