Discussing the Value of Automatic Hate Speech Detection in Online Debates

Sebastian Köffer[1], Dennis M. Riehle[1], Steffen Höhenberger[1], and Jörg Becker[1]

[1] University of Münster, European Research Center for Information Systems,
Münster, Germany
{sebastian.koeffer,dennis.riehle,steffen.hoehenberger,
joerg.becker}@ercis.uni-muenster.de

Abstract. This study discusses the potential value of automatic analytics of German texts to detect hate speech. In the course of a preliminary study, we collected a dataset of user comments on news articles, focused on the refugee crisis in 2015/16. A crowdsourcing approach was used to label a subset of the data as hateful and non-hateful to be used as training and evaluation data. Furthermore, a vocabulary was created containing the words that are indicating hate and no hate. The best performing combination of feature groups was a Word2Vec approach and Extended 2-grams. Our study builds upon previous research for English texts and demonstrates its transferability to German. The paper discusses the results with respect to the potential for media organizations and considerations about moderation techniques and algorithmic transparency.

Keywords: Natural Language Processing (NLP), Hate Speech, Text Analytics

1 Introduction

Online debates have gone off the rails. In a much-noted piece in April 2016, The Guardian published details about user comment behavior on the newspapers' website. Many comments were "crude, bigoted, or just vile". As "xenophobia, racism, sexism and homophobia were all seen regularly", the authors called it "the dark side of Guardian comments" [1]. In Germany, the amount of abusive content on the Internet during the refugee crisis has sparked a national debate on how to deal with online hate speech. German authorities formed a task force that ultimately urged social media providers to apply tougher filtering mechanism for hateful content – an action that was also criticized as excessive political correctness and censorship [2].

Detection of abusive language in user-generated online content has become an important issue for various stakeholders [3]. For instance, it is likely that hate speech and actual hate crimes relate to each other. Benesch [4] reported that hateful language delivered in the media resulted in massive violence in Kenya before and after the elections in 2007 and 2008. Similarly, German commenters proposed a relationship between the hateful online debate on refugees and attacks on homes for asylum seekers [5]. It is apparent that the nature of online debates has changed. They are

Multikonferenz Wirtschaftsinformatik 2018,
March 06-09, 2018, Lüneburg, Germany

often characterized by ideological and extreme opinions that frequently discard facts and scientific evidence. As a result, many newspapers and magazines have started fact-checking projects but at the same time, they face a serious criticism towards journalists in particular and the media in general.

Flagging hateful contents is essential for media organizations. Ignoring the problem may lead to less user traffic on their websites and companies pulling advertisements [3]. Among journalists, there is a common sentence: "Don't read the comments" [1]. In contrast to this, news organizations and their community managers try to maintain the conversation with their readers by answering comments, fact checking and explaining journalistic methodologies. As a result, comment moderation is a major manual effort for the media organizations' community managers [6].

However, many news platforms are obviously unable to cope with this demand and have limited the possibilities to comment on articles or at least on articles about contentious topics, such as refugees, conspiracy theories, climate change, and feminism [7]. In Germany, this behavior is also enforced by German law that requires community owners to delete user-generated content immediately as soon as it is known that comments contain so-called "incitement to hatred"[1]. A survey among German newspaper editors found that about 50 percent applied restrictions to the online comment sections [8].

Given the increasing amount of user-generated comments[2], we argue that analytics will ultimately be required to check for and delete abusive content and, at the same time, curate inspiration and wisdom in the debates. Especially small media organizations might be unable to deal with an avalanche of comments after publishing articles on contentious topics. Hence, the goal of this research paper is *to investigate the potential value of automatic analytics of German texts to detect hate speech*. To this end, we aim to inform community managers when thinking about putting in place algorithmic methods to support comment moderation. We draw on existing work in the area of natural language processing (NLP) for English texts. In addition to this paper, we will make public explanations for a broader audience on our projects' website[3]. This will include the possibility to access trained datasets via application programming interfaces (API).

The paper is structured as follows. In the next chapter, we list related studies on our topic. Next, Chapter 3 describes our datasets and data collection methods. Our research methodology with regard to NLP and statistics is explained in Chapter 4. After presenting our results in Chapter 5, Chapter 6 discusses implications for research and practice. The paper concludes by presenting the limitations of our work, combined with a discussion on pathways for future research.

[1] German Criminal Code in the version promulgated on 13 November 1998, Federal Law Gazette - https://www.gesetze-im-internet.de/englisch_stgb/englisch_stgb.html#p1241

[2] For instance, the New York Times receives around 9,000 submitted comments per day [9]. The Guardian receives several tens of thousands comments every day [1].

[3] This paper build upon the "Cyberhate-Mining" research project: www.hatemining.de

2 Background

The detection of abusive content, including hate speech, is not trivial: Different dictions, a huge variety of special terms for insults, a context-specific meaning, and a lot of sarcasm in the text make the task rather difficult [3]. Nonetheless, several researchers have tried to detect abusive contents in user-generated content automatically by means of text analytics [10].

Studies tried to identify hate speech in particular. Waseem and Hovy [6] used n-grams to detect hateful content in Twitter posts. Similarly, Burnap and Williams [11] detect hate speech in Tweets using a "Bag-of-Words" approach. Warner and Hirschberg [12] aim to identify hate speech in online texts by means of website and user comment annotations combined with word-sense disambiguation. Finally, Nobata et al. [3] – a group of Yahoo Labs researchers – are using deep-learning inspired methods to detect abusive comments, including hate speech. The authors apply to set of feature exploration techniques that we overtake for our study.

3 Data

3.1 Primary Dataset

We collected user-generated comments that were publicly available on news platforms on the Internet. The extraction of data included mainstream journalistic news websites as well as websites of so-called alternative media. Since most German news platforms do not offer an API to collect the comments programmatically, we used web scraping technologies. For the implementation of web scraping techniques, we used a Python framework called Scrapy[4], which has been regularly used for data collection in research projects, e.g., [13].

To select appropriate news platforms for our data collection, we rated 41 news platforms using the following criteria:

- *Comments allowed:* Platforms were excluded which did not allow user comments on articles related to the refugee crisis or which did not allow user comments at all.
- *Bots allowed:* We adhered to international standards on accessing websites with bots by respecting all bans specified in a robots.txt file. Therefore, any platforms disallowing access to bots were discarded.
- *Expected number of comments:* Since we focused on collecting large amounts of comments, we only considered platforms with a reasonable amount of articles and, even more important, with a reasonable amount of user-generated comments.
- *Estimated complexity:* We evaluated all platforms regarding their complexity. For instance, web scraping becomes more complex if websites make use of dynamic web technologies, such as AJAX or JavaScript. We followed a "low-hanging fruits" approach, starting with platforms, where comments were rather easy to scrape.

[4] https://www.scrapy.org/

Out of the 41 platforms, 16 did not allow comments to articles related to the refugee crisis. Further, two platforms denied to scrape their content using a robots.txt file. Seven of the remaining platforms did not have an active community, i.e., only very few comments could be found. And last, three platforms were discarded because of difficulties regarding their use of DDoS protection mechanisms or comments being loaded asynchronously with JavaScript. Thus, our dataset comprises 13 platforms.

Since website structures tend to vary a lot, it is necessary to implement an individual scrape mechanism for each platform. Three different scraping methods were used to ensure that only articles related to the refugee crisis were selected:

- Take articles from *topic pages*, which only list articles related to the refugee crisis. Some platforms offered special dossiers on the refugee crisis.
- Use of the news platforms' *search* function. Sometimes, this is forbidden by means of the robots.txt and then was discarded.
- Searching the websites using 79 keywords within the articles. Exemplary keywords were *asylum seeker*, *immigrant*, *integration*, *refugee*.

In total, 376,143 comments and 21,740 articles have been collected. Table 1 shows that the number of articles per platform varies greatly from 210 for "Alles Schall und Rauch" to 5,812 for "Zeit" with an average number of 1,672 articles per platform. As regards the comments, the amount per platform also varies greatly. With 182,625 comments, the platform "Welt" has by far the most comments, followed by "Focus" with 75,857 comments. All other platforms have less than the average number of comments per platform which is 28,933 comments.

Table 1. Primary dataset and scraping method

Platform	*Method*	*Articles*	*Comments*
Alles Schall und Rauch	Keyword	210	4,617
Cicero	Keyword	260	4,415
Compact	Search	328	11,764
Contra Magazin	Search	543	7,984
Epoch Times	Search	4,584	27,497
Focus	Keyword	3,959	75,857
Freie Welt	Search	1,944	13,628
Junge Freiheit	Keyword	333	2,745
NEOPresse	Search	626	11,054
Rheinische Post	Topic page	991	3,678
Tagesspiegel	Topic page	229	4,478
Welt	Keyword	1,921	182,625
Zeit	Topic page	5,812	25,792
Total		21,740	376,143

We also collected additional meta information for articles and comments, including the date of publication. Peaks are visible in late summer of 2015 and shortly after

New Year's Eve in January and February of 2016. This development corresponds to substantial events that occurred during the refugee crisis.

3.2 Evaluation Dataset

An important part of this study was to classify collected comments and to determine whether they are perceived as hateful or not. To gather a substantial collection of ratings, we collected ratings via an online survey. Using Crowdsourcing to obtain labeled training data is a common approach in research projects that deal with natural language processing to detect emotions in texts [3, 14].

Inspired by previous work on detecting hateful speech [3, 6], we used a binary categorization, so that study participants rated comments as "hate" or "no hate". In addition, study participants could also decide to skip a comment if they were unsure whether it contained hate or not.

From May to June 2016, study participants rated randomly selected comments on the project website. The selection of comments ensured that we got a similar amount of labeled data for each platform. Each comment needed to be rated by multiple participants before a final scoring decision was taken. Thus, a comment was labeled as "hate" only, if there were three hate ratings and at most one "no hate" rating and vice versa. In addition, comments that were skipped two times more than they were rated, or comments that received a 2:2 rating, were discarded.

Throughout the whole time span of our study, we received 11,973 ratings from 247 individual users in total. Among these, there were 3,875 hate, 6,073 no hate, and 2,025 unclassified ratings. According to the rules described above, this led to 2,983 labeled comments in total as depicted in Table 2. With 50 %, the largest amount of comments perceived as hate was found on "Contra Magazin", while lowest amount of perceived hate was found on "Tagesspiegel" (11 % of all comments).

Table 2. Evaluation data overview (scores) per platform

Platform	*# Hate*	*# No Hate*	*# Unclassified*	*% Hate*
Alles Schall und Rauch	54	119	61	23
Cicero	46	122	42	22
Compact	68	121	41	30
Contra Magazin	117	75	42	50
Epoch Times	93	113	39	38
Focus	45	147	57	18
Freie Welt	90	88	42	41
Junge Freiheit	74	91	47	35
NEOPresse	46	111	53	22
Rheinische Post	59	108	42	28
Tagesspiegel	26	170	40	11
Welt	55	136	48	23
Zeit	28	154	48	12
Total	801	1555	602	27.15

The overall share of hateful comments (27.15 %) is comparable high. The datasets used by Nobata et al. [3] only contain about 10 % of abusive comments. Several aspects might have contributed to the high share of hateful comments. For instance, our selection of comments is limited to articles on the refugee crisis that triggered very emotional debates. Also, our demographics of our survey participants are biased towards young people. During the rating process, all participants were asked to submit their gender age voluntarily. Out of the 247 participants, 169 did provide their age and gender; the remaining 78 users submitted neither age nor gender. The users' demographic structure is depicted in Table 3.

Table 3. Demographics of rated comments

Age group	*Male*	*Female*	*Total*
Below 25	24.7%	8.2%	33.0%
25-30	28.6%	12.7%	41.4%
31-35	5.2%	7.7%	12.9%
Over 35	6.5%	6.3%	12.7%
Total	65.0%	35.0%	100.0%

4 Methodology

4.1 Research Approach

The adoption of algorithms for comment moderation challenges the norms of transparency in journalism [15]. Originally, analytical methods to detect sentiments used vocabularies that contain sentiment words assigned with particular emotions and opinions [16]. One advantage of vocabularies is that their functioning is more comprehensible also for non-technical people.

Our study builds upon previous work by Nobata et al. [3] who evaluated several classification methods of NLP features to detect abusive content. Furthermore, we were inspired by a Kaggle competition on predicting online movie ratings from review texts [17]. Similarly to the competition, we juxtapose the vocabulary-based approach with deep-learning inspired methods that focus on the meaning of words. Most NLP studies for detecting emotions in user-generated content are using English texts only. Nobata et al. [3] note that "it remains to be seen how our approach […] would fare in other languages" (p. 152). Our study shall contribute to transfer efforts of NLP techniques with respect to German language.

4.2 Feature Extraction

We overtook the feature classification from Nobata et al. [3] who grouped their features into n-grams, linguistics, and distributional semantics (Word2Vec, Doc2Vec). In addition, we use a bag-of-words model to create a vocabulary of hateful and non-hateful words. Besides its simplicity, n-gram techniques have produced good and effective results. Thus, we decided to develop an additional feature group named

“Extended n-grams” that combines n-grams and distributional semantics. In the following, we describe the extracted feature groups in more detail[5].

Bag-of-words. To build up our vocabulary we first removed or substituted special characters, such as ä, ö, ü, and ß. Subsequently, a stop word list[6] was used to remove words from the vocabulary that are insignificant for hate speech. We also considered stemming and lemmatization for preprocessing using algorithms from the Snowball[7] project. To get numeric representations for our classifiers, we used the inverse document frequency (tf-idf). This approach yielded slightly better results than the CountVectorizer that was used in the tutorial for the Kaggle competition [17].

N-grams. We used character 2- and 3-grams. Regarding the German alphabet with 26 letters, the special characters ä, ö, ü, ß, and the space character, we obtain at most 31^2 (31^3) different 2-grams (3-grams). We used the normalized tf-idf value to determine the relative importance in the text corpus.

Linguistics. We extracted 20 features with comparatively low computational complexity. Exemplary features include the count of words, sentences, capital letters, punctuation (!?.,”), smileys, and URLs as well as the average word length and the average number of words per sentence. In order to ensure comparability between comments of different length, features were scaled in relation to the appropriate metric of the comment, i.e., number of sentences, words, characters.

Word2Vec / Doc2Vec. We used the 376,143 collected comments as training data for the Word2Vec model [18]. For feature extraction, we first transformed each word that appeared at least two times in the training into its vector representation. Subsequently, we determined the mean vector of all word vectors which is used as inputs for the features. The number of features is determined by the dimensionality of the vector. Here, we followed Nobata et al. [3] to select 50 dimensions. Similarly, we trained the Doc2Vec [19] model with all collected comments. Then, the trained model returned vector representations for all new comments. Again, we used 50 dimensions for the size of the Doc2Vec vector.

Extended n-grams. N-grams techniques cannot consider semantically equivalent but syntactically divergent texts. For instance, the words “Merkel” and “Bundeskanzlerin” most likely have a similar meaning, but the related n-grams are rather different. Our extended n-grams make use of the Word2Vec model to enrich original comment texts with nearest neighbors that are derived from the word vector representations (cosine similarity). To this end, we determined the normalized tf-idf value for each word except stop words. The higher the tf-idf measure, the more words were appended to the original comment for emphasizing words. The extended comments were then used to derive the n-gram feature.

[5] For a detailed explanation of feature extraction approaches, please refer to Nobata et al. [3].

[6] The list is available as package of the Python Natural Language Processing Toolkit (NLTK) via https://pypi.python.org/pypi/stop-words. It is maintained by Alireza Savand.

[7] A collection of stemming algorithms for several languages: http://snowballstem.org

4.3 Supervised Learning

The numerical features of the distinct feature groups (created only from the comment text itself) served as input for the classification models. For this task, only labeled comments (811 hate, 1,561 no hate) were considered. These were applied on logistic regression and evaluated to identify the best classification model[8]. The implementation was performed in Python using packages of the scikit-learn[9] module. A train and test set validation approach was chosen using a split of 75:25 between train and test set. Furthermore, we used undersampling to have equal sample size for the two classes. Thus, only 811 non-hateful comments were sampled, and the complete evaluation dataset was composed of 811 + 811 = 1,622 comments.

5 Results

Table 4 depicts our results. We report accuracy (ACC) and F-score for our models. We also tried whether combinations of two feature groups perform better. The bag-of-words approach obtained the best ACC value with 67.8 percent. The highest F-score was obtained using the Word2Vec with 0.67. The best performing combination of feature groups was Word2Vec and Extended 2-grams (ACC = 0.7068, F-score= 0.70).

Table 4. Performance of feature groups for classification task

Feature group	*ACC*	*F-score*
Bag-of-words	0.6780	0.51
2-grams	0.6206	0.64
3-grams	0.6551	0.65
Linguistics	0.5689	0.53
Word2Vec	0.6650	0.67
Doc2Vec	0.6477	0.63
Extended 2-grams	0.6009	0.61
Extended 3-grams	0.6059	0.61

For the bag-of-words approach, Table 5 shows the words that are most indicative for hateful comments, i.e., the appearance of the word “Europe” mostly increases the probability that a comment is considered as hateful.

Some of the hate indicative words can be related to political topics. For instance, chancellor Merkel promoted an open culture for refugees and faced a lot of criticism in online debates. The list of the non-hateful indicative words contains auxiliary words (gar, vielen) that could have been part of the stop word list. However, for the purpose of this study, we stick to the lists that we overtook from previous work.

[8] We also applied Support Vector Machines. Since the results were similar but slightly worse compared to the logistic regression, we do not report the figures as part of this paper.

[9] https://www.scikit-learn.org

Table 5. Five most indicative words for hateful and non-hateful comments[10]

Hate		*No hate*	
German	*English*	*German*	*English*
europa	Europe	finde	find
verbrecher	criminals	artikel	article
luegen	lies	integration	integration
duerfen	may / can	vielen	many
merkel	merkel	gar	even

6 Discussion

In this study, we examined the value of text analytics for an automatic detection of hate speech in German texts. Therefore, we conducted a preliminary study in which we collected a dataset of user comments on German news articles, focused on the refugee crisis in Germany in 2015/16. A crowdsourcing approach was used to label a subset of the data to be used as a training and evaluation dataset. We then selected feature groups that are anchored in related other scientific work to evaluate a classification model using a logistic regression approach. Furthermore, a vocabulary has been created containing the words that are indicating hate and no hate.

Our study demonstrates that previously used concepts by other researchers [3, 6, 17] can be transferred to German texts. However, German language specifics, like irregular plural forms, compound nouns or anglicisms complicate the process of stemming and lemmatization. As a final result, we achieved best results with an accuracy of approximately 70 % and an F-score of 0.7. Thus, our results are slightly outperformed by recent academic work that used similar methods with English texts[11]. Given the limitations of our work, particularly the available datasets, we rate the preliminary study's outcome as promising and satisfactory.

Our results partly confirm the strength of character-based NLP techniques. Despite their simplicity, character-based (and also word-based) techniques do not perform considerable worse in our study than more complicated mechanisms, such as distributional semantics. However, we think that distributional semantics and other deep-learning inspired techniques have the potential to outperform character-based and word-based techniques as soon as the datasets are big enough. For instance, training Word2Vec on a lot more text should improve performance [17].

As regards our vocabulary, we argue that our list of most indicative words may contribute to an increased transparency of analytical methods. Algorithmic methods for comment moderation are soon criticized as automatic censorship to repress political opponents [15]. If analytical approaches are able to share intermediate results and explanations, they may have the potential to be more comprehensive and more

[10] For the purpose of this paper, we translated the words into English.

[11] Nobata et al. [3] achieved F-scores up to 0.81 to detect abusive content in a news dataset by combining similar feature groups that are used in this study. Waseem and Hovy [6] reached F-scores up to 0.74 using character n-grams to detect hateful comments in Tweets.

objective than any netiquette that is used as a guideline for manual comment moderation, which usually happens behind the scenes. However, while our hate-indicative word lists may slightly open the analytical black box, they may also be a target for criticism itself if people do not agree with certain elements of the list.

7 Limitations and Outlook

First and foremost, the biggest shortcoming of this study is the relatively small size of the dataset. To train our algorithms, we were confined to a set of 2,372 labeled comments. Related studies that apply machine learning with NLP use massively bigger datasets with hundreds of thousands labeled texts [3, 20]. We plan to further increase our labeled dataset in the future and are confident that this will increase the chances to obtain the same evaluation scores like other researchers. This is particularly important to gain more acceptances for algorithmic approaches by journalists and news organizations.

Second, although web scraping is the (only) suitable approach for us, it has several pitfalls. For instance, we cannot guarantee that our scraping method has collected all relevant articles on the refugee crisis or whether there have been errors when scraping the comment texts. Furthermore, we had to discard many news platforms that would have been worth to analyze. However, our scripts worked reliable so that we were able to obtain the data rather easily. Nevertheless, the fact that larger international newspapers (e.g., New York Times) offer APIs might encourage German news platforms to follow at some point.

Third, our study is limited to comments on news platforms and articles on the refugee crisis. Thus, its findings cannot be transferred directly to other topics and platforms, such as social media platforms. We chose this focus because we believe that media organizations will ultimately need analytical approaches to maintain online debates on their websites. If online debates continue to move from journalistic media to social media platforms, journalists will lose their opportunity to steer and enrich the debates, and ultimately be ever more dependent on social media platforms [21]. To work on bigger datasets and better data labeling, we encourage German media organizations and researchers to join forces. The Coral Project[12] where New York Times, Washington Post, developers, and researchers team up to “build better communities around their journalism” is the prime example.

Fourth, our dataset is already pre-filtered by the news platforms that use very different moderation strategies to delete hateful comments before they are publicly visible. We do not precisely know which semi-automatic techniques for comment moderation are already in place[13]. It would be interesting to have access to the raw data which is likely to contain more hateful contents. Hence, the results must be interpreted carefully, because the dataset does not directly represent what people write in the online comment sections.

[12] The Coral Project: https://www.coralproject.net

[13] The Guardian revealed that 1.4 million (2% of the total) comments had been blocked by February 2016 using manual moderation [1].

Fifth, the study's participants to label the comment texts as hateful or non-hateful are not representative of the whole population. They were recruited via social media platforms among people in a University context. Hence, participants are rather young with a relatively high level of education. Furthermore, it is unlikely that many participants possess journalistic expertise or experience with community moderation. Future studies should use a more representative sample of the population.

Sixth and finally, our decision to use a binary classification between hate and non-hate is problematic, since every individual might have a different understanding what hate means. In a subsequent study, we plan to further detail the comment ratings to be able to distinguish between different aspects of hate speech such as insults, xenophobia, and threats.

To conclude, an analytical tool for comment moderation must deliver a high level of accuracy to meet high journalistic standards. Accuracy values around 70 % as in our preliminary study or around 80 % like in related studies are probably still insufficient. But even if better datasets and algorithms allow better prediction rates, they do not necessarily call for an automatic deletion of hateful comments. Since no analytical approach is likely to guarantee almost zero failure in a foreseeable future, false positives may continue to trigger discussions of undesirable censorship by media. From the discussions of our study results with several stakeholders we conclude that semi-automatic approaches, where moderators review the analytical outcomes are more feasible. Such approaches can also include the commenters themselves since they could get immediate feedback about the submitted comment text. This example is just a fraction of potential pathways that can be envisaged through the use of analytical methods. We hope that our paper contributes to enabling the use of text analytics to bring online debates back on track – pursuing fruitful and enriching discussion on the web. It is an effort worth making.

Acknowledgements

This paper was written in the context of the research project “Cyberhate-Mining” (www.hatemining.de). We gratefully acknowledge the constructive comments and cooperation provided by the participating students, advisors, and colleagues.

References

1. Gardiner, B., Mansfield, M., Anderson, I., Holder, J., Louter, D., Ulmanu, M.: The dark side of Guardian comments, https://www.theguardian.com/technology/2016/apr/12/the-dark-side-of-guardian-comments.
2. Faiola, A.: Germany springs to action over hate speech against migrants, https://www.washingtonpost.com/world/europe/germany-springs-to-action-over-hate-speech-against-migrants/2016/01/06/6031218e-b315-11e5-8abc-d09392edc612_story.html.

3. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive Language Detection in Online User Content. In: International World Wide Web Conference Committee (IW3C2). Montreal, Quebec, Canada (2016).
4. Benesch, S.: Countering dangerous speech to prevent mass violence during Kenya's 2013 elections. (2014).
5. Lobo, S.: Wie aus Netzhass Gewalt wird und was dagegen hilft, http://www.spiegel.de/netzwelt/netzpolitik/netzhass-und-gewalt-was-man-dagegen-tun-kann-lobo-kolumne-a-1048799.html.
6. Waseem, Z., Hovy, D.: Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: NAACL Student Research Workshop. San Diego, CA, USA (2016).
7. The Coral Project Community: Shutting down onsite comments: a comprehensive list of all news organisations, https://community.coralproject.net/t/shutting-down-onsite-comments-a-comprehensive-list-of-all-news-organisations/347.
8. Siegert, S.: Nahezu jede zweite Zeitungsredaktion schränkt Online-Kommentare ein, http://www.journalist.de/aktuelles/meldungen/journalist-umfrage-nahezu-jede-2-zeitungsredaktion-schraenkt-onlinekommentare-ein.html.
9. Etim, B.: The Most Popular Reader Comments on The Times, http://www.nytimes.com/2015/11/23/insider/the-most-popular-reader-comments-on-the-times.html.
10. Schmidt, A., Wiegand, M.: A Survey on Hate Speech Detection using Natural Language Processing. Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Valencia, Spain. 1–10.
11. Burnap, P., Williams, M.L.: Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy and Internet Vol 7(2), pp. 223–242 (2015).
12. Warner, W., Hirschberg, J.: Detecting Hate Speech on the World Wide Web. In: Proceedings of the 2nd Workshop on Language in Social Media, pp. 19–26. Montréal, Canada (2012).
13. Wang, J., Guo, Y.: Scrapy-based crawling and user-behavior characteristics analysis on Taobao. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover,. pp. 44–52 (2012).
14. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon. In: Computational Intelligence, pp. 436–465 (2013).
15. Diakopoulos, N., Koliska, M.: Algorithmic Transparency in the News Media. Digital Journalism. (2016).
16. Liu, B.: Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies Vol. 5, pp. 1–167 (2012).
17. kaggle.com: Bag of Words Meets Bags of Popcorn, https://www.kaggle.com/c/word2vec-nlp-tutorial.
18. Mikolov, T., Corrado, G., Chen, K., Dean, J.: Efficient Estimation of Word Representations in Vector Space. International Conference on Learning Representations (ICLR 2013), pp. 1–12 (2013).
19. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. International Conference on Machine Learning - ICML 2014, pp. 1188–1196 (2014).
20. Rao, Y., Lei, J., Wenyin, L., Li, Q., Chen, M.: Building emotional dictionary for sentiment analysis of online news. World Wide Web Vol. 17, pp. 723–742 (2014).
21. Brodnig, I.: Der Hass im Netz – und was dagegen zu tun ist, http://www.carta.info/81616/hass-als-instrument/.