# Context-aware Classification of News Comments

Johannes Filter
Hasso Plattner Institute, University Potsdam
Potsdam, Germany
Johannes.Filter@student.hpi.de

## KEYWORDS

Online News Comments, User-generated Content, Natural-Language Processing, Machine Learning

## 1 INTRODUCTION

Online news outlets are drowning in the vast number of user comments. While there are certainly high-quality comments that e.g. give a new perspective to a story, there are as well comments which are rather unwelcome. They range from out-of-topic discussion to the use of offensive language. A lot of websites are closing their comment sections because they cannot cope with the sheer amount of user-generated content. News organizations already have economic difficulties[1] and they cannot afford to moderate comments. But with the help of new natural-language processing (NLP) and machine learning techniques, there is hope to automatically analyze user comments. Which as a consequence, frees up resources from moderation and opens the comment section again.

There already exists work to detect hate speech or abusive language [5, 23, 33] in user-generated content. In this work, we want to take a different perspective on the issue. Instead of eliminating 'bad' comments, we want to identify high-quality comments. We are doing this by classifying after specific criteria that constitutes those comments. Such criteria are for instance whether a comment is on the topic of the article or if supports its claims with an argument. In contrast to previous machine learning work, we assume that the article as a context is important to consider for making classifications. The formal definition is that given a training set $S = \{(X_n, A_n, y_n)\}_{n=1}^{N}$ of triple-wise data, where $X \in \chi$ is the comment text, $A \in \alpha$ is the article text and $y \in Y$ is its corresponding label.

In addition, we present a method of normalizing user votes to derive a comparable popularity score for each comment. We assume that the popularity of a comment is a rough proxy for its quality. Nevertheless, are we aware that the vote of a large number of people does not necessarily lead to justified judgments as seen for instance in the Brexit referendum. But we still assume that, especially in an enormous corpus, the vote of the crowd has some kind of meaning. News aggregators such as Hacker News[2] or Reddit[3], live from their user votes to identify high-quality content. Still, there are a lot of factors that come into play when people vote and our method only operates on the number of votes at one specific moment in time. We neither know who casts these votes nor when. This may not be enough to come to a reasonable, objective and comparable popularity score. We will critical investigate our method on multiple existing comment datasets before drawing conclusions.

But what is the bigger picture of user comments? Why exactly is it important to have them? Glenn Greenwald[4] states:

> "Journalists often tout their responsibility to hold the powerful accountable. Comments are a way to hold journalists themselves accountable."

So one facet is about controlling the journalists themselves. Before the Internet, people could only send letters to the editors. Part of those letters was printed but those comments were certainly moderated strictly. So the introduction online news comments were an emancipatory act of depriving journalists of their role as gatekeepers. This also allowed to have a debate about in issuer in the wider public. This touches on the discourse ethics[5] as proposed by Jürgen Habermas into practice. The principle of discourse ethics: a group of people with a rationale debate comes to a common conclusion which manifests the morale. This is a start contract to Immanuel Kant's categorical imperative which focuses on individuals. The theoretical concept of the discourse ethics can be set in practice in the comment section albeit in some variation. In essence, it is about the belief that the collective can come to better conclusions than a sole person.

The field of journalism is tightly coupled with the technological advancement of our society. Only through the invention of the letterpress by Johannes Gutenberg could information be spread so fast. This allowed the profession to establish and flourish and the digital revolution also affects journalists. There is a long tradition of supporting their work with digital technologies. It started in the late '60s with computer-assisted reporting[6] and lead to current ideas about automatic reporting[7]. Right now, there is a larger effort by supporting newspapers in managing their user comments. On example is the unprecedented Coral Project[8], a cooperation among Mozilla, the New York Times and the Washington Post, that interviewed more than 400 experts in 150 newsrooms to develop an IT system to manage comments. In the following section, we give a detailed overview of related scientific work of natural-language processing with machine learning on news comments.

[1] https://en.wikipedia.org/wiki/Decline_of_newspapers
[2] https://news.ycombinator.com/
[3] https://www.reddit.com/
[4] https://theintercept.com/2017/12/18/comments-coral-project/
[5] https://en.wikipedia.org/wiki/Discourse_ethics
[6] https://en.wikipedia.org/wiki/Computer-assisted_reporting
[7] https://en.wikipedia.org/wiki/Automated_journalism
[8] https://coralproject.net/

## 2 RELATED WORK

The literature that is related to this work can be roughly split into two categories. One is about news comments and machine learning applied to it. And the second is about recent trends in natural-language processing (NLP) with deep learning.

### 2.1 News Comments

With the beginning of the Web 2.0, there is an abundance of user-generated content. Some influential earlier work analyzed comments on Digg[9] [9, 17], and predicting the popularity of online content on Youtube and Digg [38] or Reddit [34]. There is work done by Park et al. in the field of Human-Computer Interaction [25] where they build a system to manage comments by incorporating feature-based traditional machine learning.

Recent work in the research area of comment analysis focuses on identifying moderator-picked comments on New York Times website ('NYT Picks'). Nicholas Diakopoulos [6] present several criteria that distinguish the two classes, three of them that can be computed. Kolhatkar and Taboada [15] predict the NYT Picks with traditional, feature-based machine learning as well as deep learning. They achieved an F1 score of 0.84. In their follow-up work, they constructed the 'SFU Opinion and Comments Corpus' [16], where they labeled over 1000 comments for 'constructiveness' and predicted it with an accuracy of 72.59% [14].

Napoles et al. [21] focused on detecting 'good' discussions in the comment section. They defined 'good' discussions as ERIC: Engaging, Respectful, and/or Informative Conversation. They as well created a corpus of comments, 'The Yahoo News annotated comments corpus' [22], labeled a subset of about 10k comments and 2.4k threads and predicted ERIC threads with an F1 score of 0.73.

The work so far did not consider the context of an article. Namely, the article and also other comments. The very recent of work by Cheng et al. [4] considers the abstract of the news article as well as surrounding comments to classify comments. It adapts the text matching methods of Wang et al. [39] that uses LSTM [11] with attention mechanism. But their work has one weakness: It treats all comments with over 10 up-votes as positive and the rest as negative samples. This is a gross simplification for multiple reasons. For instance, earlier comments are more likely to get more up-votes. So they may only predict comments that appear earlier in the discussions than others. Even though they achieved an accuracy of 70.75 and an F1 score of 80.73, their true contribution is unclear. Nevertheless, their deep learning network architecture of combining comment, article and other comments can be a starting point for our work. Gao and Huang [8] report experiments where they tried to bring context to hate speech detection which improved their results.

Loosely related is the work by Qin et al. [31] who automatically generate high-quality comments. Also the problem of 'stance detection' of detecting whether a response to a statement is affirmative or opposing. Some promising work has been done by Kochkina et al. [13]. In addition, the prediction of the helpfulness of user products reviews done by Singh et al. [37] is relevant. Another work on product reviews and recommender system has been done by Zheng et al. [40]. They encoded the product description as well as reviews with an LSTM respectively before combining the two sub-networks into a classifier.

Outside of the computer science community, there exists qualitative analysis of comments that should guide our work. Loosen et al. [18] formulated several comment quality indicators after conducting interviews with news professionals as well as developing a software mockup. Among other things, they list 'comment length' and 'readability' but also the 'reference to the article' and 'references to other comments' as an indicator. Their work is built upon earlier research done by Diakopoulos et al. [7] and Noci et al. [24]. There is also an abundance of comment guidelines that outline good comments. The New York Time writes in their guidelines[10]: "We are interested in articulate, well-informed remarks that are relevant to the article." Also, the community guidelines by the Guardian requires commentators to "[k]eep it relevant"[11]. So the relation to the article plays a role when judging comments for its quality. Current machine learning methods often are too simplistic and neglect this feature alltogether.

### 2.2 Downstream NLP tasks with Language Models

Within the deep learning NLP community, there is a trend of abandoning word embeddings as popularized by Word2Vec [20], GloVe [26] or fastText [2] in favor of text representations derived from language models. There are two main reasons for this: First, the word tokens are global which means that they are context-agnostic. This is a drawback because there might be multiples meanings for a word. For instance 'bear' can describe an animal or act as a verb. The true meaning in a sentence can only be inferred by considering the surrounding words. Second, the meaning of the word gets lost after the embedding layers. Subsequent, deeper layers do not have access to the 'meaning' of a word that was injected through the word embeddings.

Language models (LM) try to predict the next word based on previous words. This is a challenging task so the model has to learn the nuances of a language over long sequences of text. But they do not require annotated data and there is an abundance of raw text freely available (e.g. Wikipedia). After training an LM, there is the hope of transforming its capabilities to other tasks. The Elmo embeddings by Peters et al. [28] were one of the first ones to use the potential of LM surpassing earlier work by McCann et al. [19] (CoVe). They use the internal state of a language model to get vectors as

[9] http://digg.com/

[10] https://help.nytimes.com/hc/en-us/articles/115014792387-Comments

[11] https://www.theguardian.com/community-standards

a building block that can directly replace traditional word embeddings. Similar work has been done by of Abkib et al. [1] for Flair[12]. A more detailed explanation about LM and their use in text representation can be found in the recent publication by Peters et al. [29].

Howard and Ruder show with Ulmfit [12] how to apply transfer learning to fine-tune from an ordinary LM to downstream classification task. They broke multiple state-of-the-art performances on text classification[13] and sentiment detection[14], e.g. on the IMDb dataset[15]. In contrast to Elmo, they provide a full architecture and methodology for applying their approach to real-world scenarios. They point out that the challenging part is the method of how to fine-tune from the LM to the task. The OpenAI transformer by Radford et al. [32] also starts with an LM and 'transforms' it to downstream tasks. Their performance for text classification is below than Ulmfit. But they provided also ways to 'transform' the LM to other tasks such as question answering. Both papers highlight, that there is a lot to explore in the field of LM fine-tuning. There is a comparison of several approaches done by Perone et al. [27] – sadly without Ulmfit or the OpenAI transformer. Transfer learning in NLP is currently a hot topic and there might be no clear answers. But doing it with LM is not the only way. Also, the 'Universal Sentence Encoder' by Cer et al. [3] makes use of transfer learning but without LMs.

# 3 CONTEXT-AWARE CLASSIFICATION OF NEWS COMMENTS

We want to draw from the recent developments in NLP to classify news comments. In addition, we will propose a model that considers the article as well when judging the comments. This is a combination of text classification and text matching. We follow the recent works of language model fine-tuning as done by Ulmfit [12] or the OpenAI Transformer [32]. We will first present several baselines before describing the model in detail.

## 3.1 Baseline

The baseline methods range from traditional feature-based approaches as described by Kolhatkar and Taboada [15] with features such as average word/sentence/comment length and using stylometric features as described by Potthast et al. [30] to well-established attention-based deep learning architectures. Further, to verify that our method actually works, we will compare use Ulmfit [12] without any relation to the comment.

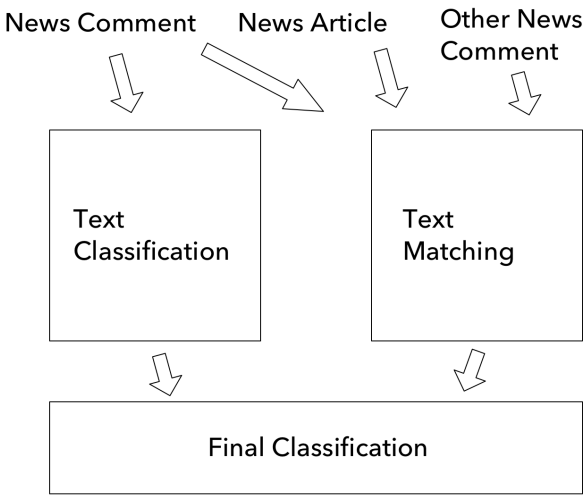


**Figure 1: Overview of the model architecture.**

## 3.2 Model

We propose a model architecture that contains two sub-networks, one only on the comment's text and one also for the text matching between article's text and comment's text. Both sub-structures are combined into a final layer to classify the comments. For the text classification, we follow the work of Ulmfit [12]. For the text matching, we follow the work of the OpenAI transformer [32]. So the general idea is that we first train a general language model on a large text corpus. Then, transform and fine-tune the trained language model to our task. An overview of the complete architecture is shown in Figure 1. The whole process of fine-tuning looks as follows:

(1) Train a language model on a large corpus (or use a pre-trained one)
(2) Fine-tune text classification module
  (a) Retrain the language model on news comments
  (b) Fine-tune text classification on news comments
(3) Fine-tune text matching module
  (a) Retrain the language model on news articles
  (b) Retrain another language model on news comments
  (c) Fine-tune text matching between news articles and news comments
(4) Train the final classifier

We will as well investigate other approaches. One way is, similar to Zheng et al. [40], to encode the comment and the article with an LSTM respectively before joining into a common fully-connected layer. Also, certain classification

[12] https://github.com/zalandoresearch/flair
[13] http://nlpprogress.com/text_classification.html
[14] http://nlpprogress.com/sentiment_analysis.html
[15] https://ai.stanford.edu/~ang/papers/acl11-WordVectorsSentimentAnalysis.pdf

problems, e.g. whether a comment is off-topic can be interpreted as a text matching problem. So complex two-stage model may actually not be needed.

### 3.3 Further Ideas

To further improve the model, we could incorporate external knowledge into our model. Some concrete information about people and e.g. their affiliation (e.g. political party membership) may not be in the learned corpora. One way would be to use a Named-Entity Recognition (NER) method and then to look up the entity on external knowledge resources such as Wikipedia/Wikidata.

## 4 DATA

In this section, we will give an overview of available partly-annotated data sources as well as a method to generate data labels for popularity out of user votes.

### 4.1 Comment Corpora

As of today, there exist four bigger news comments. Two in English, one in German and on in Chinese that are presented in Table 4.1.

| Dataset | Description | Annotations |
|---|---|---|
| SFU Opinion and Comments Corpus (SOCC) [16] [16], in English | 10k articles with their 663k comments from 303k comment threads, from 2012 to 2016, from Canadian newspapers, with up- and down-votes | 1,043 annotated comments in responses to 10 articles, labeled for constructiveness and toxicity |
| Yahoo Annotated News Corpus (YNACC) [17][22], in English | 522k comments from 140k threads posted in response to Yahoo News articles, with up- and down-votes | 9.2k comments labeled for agreement, audience, persuasiveness, sentiment, tone, (off-)topic and 2.4k threads labeled for agreement, constructiveness, type |
| One Million Posts Corpus (OMPC) [35, 36] [18], in German | 12k articles, 1M comments, from an Austrian Newspaper, with up- and down-votes | 11k comments with the following labels: sentiment, off-topic, inappropiate, discriminating, feedback, personal studies, argument used |
| Tencent News Corpus (TNC) by Qin et al. [31], in Chinese | 200K articles and 4.5M comments, from a Chinese news website, 2017, with up-votes | 40k comments labeled for quality (from 1 to 5) |

For this work, the YNACC for English and the OMPC for German are especially interesting because of their fine-grained annotations. The number of annotated comments in SOCC are too few. TNC has enough data but only one label of quality. YNACC and OMPC have also some overlapping annotation criteria: Off-topic and sentiment.

### 4.2 Turning User Votes into Classes

Although there exists some labeled data, the amount is still relatively small. Halevy et al. [10] showed that more data, in general, helps to improve the performance of deep learning models. So we propose a method to generate binary labels

of popularity out of user votes. This method helps to create more data and as a consequence, hopefully, better performance. In the following are the basic steps:

(1) Remove non-root comments
(2) Remove articles with few comments
(3) Remove articles with few up-votes
(4) Rank comments by chronological order
(5) Only consider first N comments per article
(6) Calculate relative up-votes for each comment
(7) Classify the comments with the most up-votes as positive and the least as negative, leaving out the middle

One idea is that do not take the up-votes directly but normalize them first. A second thought is to filter out the comments where there is no clear evaluation of 'the crowd'. This is done by ordering the comments by normalized up-votes and filtering out comments in the middle. There is the hope that the really popular ones are good and the really unpopular ones are bad. This hypothesis has to be verified which can be done by applying the methods to datasets within the Table 4.1. A further modification is required for comments that have up-votes as well as down-votes. For this two additional classes: 'controversial' and 'boring' but how exactly the comments are assigned to a class is due to further research.

# 5 EVALUATION

Because our contributions are two-fold, we have two different sub-sections for the evaluation.

## 5.1 Machine Learning

To verify the performance of our proposed architectures as described in Section 3, we compare it to previously reported results. The authors of OMPC report baselines [35] for all classification criteria that we want to challenge. The criteria 'off-topic' with a prediction of 0.2579, recall of 0.6241 and an F1 score of 0.3516 leaves room for improvement. Because our model has the relation between comment and article, it ought to achieve better results. But also the majority of the other criteria have F1 scores ranging from 0.5 to 0.7 that show these problems are far from being solved. (There will be a paper from Hamburg available shortly, where they also work on the OMPC and report F1 scores). Unfortunately, there are no reported values for the comments in YNACC only per thread level. For SOCC, the authors [14] report an accuracy of 0.7259 for identifying 'constructive' comments. Since their definition constructive implies being relevant to the article, we have to be better as well to prove our method actually works.

## 5.2 Turning User Votes into Classes

To evaluate our method of normalizing the user votes proposed in Section 4.2, we compare the resulting classification against the golden truth (the annotated parts) of the comments in Table 4.1. There are certain labels such as 'constructiveness', 'argument used', or 'on-topic' that should occur significantly more often in our 'popular' comments. In a second step, we conduct a user study ($N^{19}$=10) with a quantitative and a qualitative part to determine whether the 'popular' comments are actually superior to other comments. First, participants are required to assign quality labels to comments after their read the corresponding article. Participants are asked to read three articles and judge ten comments per article. Second, in a semi-structured interview, we will try to elicit additional information about their decisions to get an understanding of what a good comment is in particular and how it relates to user voting on comments.

# REFERENCES


[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. [n. d.]. Contextual String Embeddings for Sequence Labeling. ([n. d.]), 12.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[4] D. Chen, S. Ma, P. Yang, and X. Sun. 2018. Identifying High-Quality Chinese News Comments Based on Multi-Target Text Matching Model. *ArXiv e-prints* (Aug. 2018). arXiv:cs.CL/1808.07191

[5] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM '17)*. 512–515.

[6] Nicholas Diakopoulos. 2015. Picking the NYT Picks: Editorial criteria and automation in the curation of online news comments. (2015).

[7] Nicholas Diakopoulos and Mor Naaman. 2011. Towards Quality Discourse in Online News Comments. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work (CSCW '11)*. ACM, New York, NY, USA, 133–142. https://doi.org/10.1145/1958824.1958844

[8] Lei Gao and Ruihong Huang. 2017. Detecting Online Hate Speech Using Context Aware Models. *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning* (Nov 2017). https://doi.org/10.26615/978-954-452-049-6_036

[9] Vicenç Gómez, Andreas Kaltenbrunner, and Vicente López. 2008. Statistical Analysis of the Social Network and Discussion Threads in Slashdot. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. ACM, New York, NY, USA, 645–654. https://doi.org/10.1145/1367497.1367585

[10] Alon Halevy, Peter Norvig, and Fernando Pereira. 2009. The unreasonable effectiveness of data. *IEEE Intelligent Systems* 24, 2 (2009), 8–12.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. *arXiv:1801.06146 [cs, stat]* (Jan. 2018). http://arxiv.org/abs/1801.06146 arXiv: 1801.06146.

[13] Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at SemEval-2017 Task 8: Sequential approach to rumour stance classification with branch-LSTM. *arXiv preprint arXiv:1704.07221* (2017).

[14] Varada Kolhatkar and Maite Taboada. 2017. Constructive language in news comments. In *Proceedings of the First Workshop on Abusive Language Online*. 11–17.

[15] Varada Kolhatkar and Maite Taboada. 2017. Using New York Times Picks to Identify Constructive Comments. Association for Computational Linguistics, 100–105. https://doi.org/10.18653/v1/W17-4218

[16] Varada Kolhatkar, Hanhan Wu, Luca Cavasso, Emilie Francis, Kavan Shukla, and Maite Taboada. 2018. The SFU Opinion and


[19] number of participants

Comments Corpus: A Corpus for the Analysis of Online News Comments. (2018).

[17] Cliff Lampe and Paul Resnick. 2004. Slash(Dot) and Burn: Distributed Moderation in a Large Online Conversation Space. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 543–550. https://doi.org/10.1145/985692.985761

[18] Wiebke Loosen, Marlo Hring, Zijad Kurtanovi, Lisa Merten, Julius Reimer, Lies van Roessel, and Walid Maalej. 2017. Making sense of user comments: Identifying journalists requirements for a comment analysis framework. *Studies in Communication | Media* 6, 4 (2017), 333–364. https://doi.org/10.5771/2192-4007-2017-4-333

[19] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. 6294–6305.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781 http://arxiv.org/abs/1301.3781

[21] Courtney Napoles, Aasish Pappu, and Joel R Tetreault. 2017. Automatically Identifying Good Conversations Online (Yes, They Do Exist!).

[22] Courtney Napoles, Joel Tetreault, Aasish Pappu, Enrica Rosato, and Brian Provenzale. 2017. Finding good conversations online: The Yahoo News annotated comments corpus. In *Proceedings of The 11th Linguistic Annotation Workshop*. 13–23.

[23] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 145–153. https://doi.org/10.1145/2872427.2883062

[24] Javier Díaz Noci, David Domingo, Pere Masip, JL Micó, and C Ruiz. 2012. Comments in news, democracy booster or journalistic nightmare: Assessing the quality and dynamics of citizen debates in Catalan online newspapers. In *International Symposium on Online Journalism*, Vol. 2. 46–64.

[25] Deokgun Park, Simranjit Sachar, Nicholas Diakopoulos, and Niklas Elmqvist. 2016. Supporting Comment Moderators in Identifying High Quality Online News Comments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1114–1125. https://doi.org/10.1145/2858036.2858389

[26] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

[27] C. S. Perone, R. Silveira, and T. S. Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *ArXiv e-prints* (June 2018). arXiv:cs.CL/1806.06259

[28] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[29] Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wentau Yih. 2018. Dissecting Contextual Word Embeddings: Architecture and Representation. *arXiv:1808.08949 [cs]* (Aug. 2018). http://arxiv.org/abs/1808.08949 arXiv: 1808.08949.

[30] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein. 2017. A Stylometric Inquiry into Hyperpartisan and Fake News. *ArXiv e-prints* (Feb. 2017). arXiv:cs.CL/1702.05638

[31] L. Qin, L. Liu, V. Bi, Y. Wang, X. Liu, Z. Hu, H. Zhao, and S. Shi. 2018. Automatic Article Commenting: the Task and Dataset. *ArXiv e-prints* (May 2018). arXiv:cs.CL/1805.03668

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[33] Julian Risch and Ralf Krestel. 2018. Delete or not Delete? Semi-Automatic Comment Moderation for the Newsroom. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. 166–176.

[34] Georgios Rizos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2016. Predicting News Popularity by Mining Online Discussions. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 737–742. https://doi.org/10.1145/2872518.2890096

[35] Dietmar Schabus and Marcin Skowron. [n. d.]. Academic-Industrial Perspective on the Development and Deployment of a Moderation System for a Newspaper Website. ([n. d.]), 4.

[36] Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One Million Posts: A Data Set of German Online Discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 1241–1244. https://doi.org/10.1145/3077136.3080711

[37] Jyoti Prakash Singh, Seda Irani, Nripendra P. Rana, Yogesh K. Dwivedi, Sunil Saumya, and Pradeep Kumar Roy. 2017. Predicting the helpfulness of online consumer reviews. *Journal of Business Research* 70 (2017), 346 – 355. https://doi.org/10.1016/j.jbusres.2016.08.008

[38] Gabor Szabo and Bernardo A. Huberman. 2010. Predicting the Popularity of Online Content. *Commun. ACM* 53, 8 (Aug. 2010), 80–88. https://doi.org/10.1145/1787234.1787254

[39] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. *arXiv:1702.03814 [cs]* (Feb. 2017). http://arxiv.org/abs/1702.03814 arXiv: 1702.03814.

[40] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 425–434. https://doi.org/10.1145/3018661.3018665