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, NLP, Machine Learning

1 INTRODUCTION

Online news outlets are drowning in the vast number of user comments. While there a certainly high-quality comment that e.g. give a new perspective to a story, there are as well comment 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 section because they cannot cope with the sheer amount. 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 [4, 20, 29] 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 classify comments for arbitrary classification criteria. Such criteria are for instance whether a comment is on topic or if it contains an argument. By this, so we circumvent this sharp binary distinction between 'bad' and 'good' and make the classification more fine-grained. In addition, we present a method of normalizing user votes to derive the popularity of a comment. We assume that the popularity is a rough proxy for the quality of a comment we will as well investigate the hypothesis in this work as well.

From a machine learning perspective, we want to predict the class of a comment not only on its text but also its relation to the news article as well as other comments. This can be formulated as instance probabilities as follows: $p(X_k|A|C|C_O)$. Where A is the article, C is the comment, C_O are other comments, and k are possible outcomes of classes X_k.

But why exactly is it important to have user comments at all? Glenn Greenwald[2] 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[3] 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 manifest 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.

There is a long tradition of supporting journalistic work with digital technology. It started in the late 60s with computer-assisted reporting[4] and leading to current ideas about automatic reporting[5]. Right now, there is a larger effort by supporting newspapers in managing their user comments. There is the unprecedented Coral Project[6], 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.

2 RELATED WORK

The literature that is related to this work can be roughly split into two categories. One is about news comments and the second is about recent trends in 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[7] [7, 15], and predicting the popularity of online content on Youtube and Digg [34] or Reddit [30]. There is work done by Park et al. in the field of Human-Computer Interaction [22] 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 [5] present several criteria that distinguish the two classes, three of them that can be computed. Kolhatkar and Taboada [13]

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

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' [14], where they labeled over 1000 comments for 'constructiveness' and predicted it with an accuracy of 72.59% [12].

Napoles et al. [18] 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' [19], 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. [3] 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. [35] that uses LSTM [9] with attention mechanism. But their work has one weakness: It treats all comments with over 10 upvotes 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 upvotes. So the true contribution of their work is unclear.

Loosely related is the work by Qin et al. [27] 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. [11]. In addition, the prediction of the helpfulness of user products reviews done by Singh et al. [33] is relevant. Another work on product reviews and recommender system has been done by Zheng et al. [36]. They encoded the product description as well as reviews with an LSTML respectively before combining the 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. [16] formulated several comment quality indicators after conducting several interviews with News professionals. Earlier work by Diakopoulos et al. [6] and Noci et al. [21] highlight the quality of comments. There is also an abundance of comment guidelines that outline good comments. The New York Time writes in their comment guidelines[8]: "We are interested in articulate, well-informed remarks that are relevant to the article."

2.2 Downstream NLP tasks with Language Models

Within the deep learning NLP community, there is trend of abandoning word embeddings as popularized by Word2Vec [17], GloVe [23] or fastText [2] in favor of text representation derived from language models[9]. There are two main reasons for this: the vector tokens are global which means that they are context-agnostic. This is a drawback because the meaning of a word may change depending on the surround words. And second, when used in deep learning models, when using word embedding layer, the meaning of the word only gets used in the first layer. All deeper layers do not have access to the meaning.

Why and how can language models help? Language predicts the next word based on previous words. They are trained unsupervised and there is an abundance of text freely available (e.g. Wikipedia). After training the language model, it can be used for other tasks. The Elmo embeddings by Peters et al. [25] where one of the first ones to use this idea. Using the internal state of a language model to go vector representation for downstream tasks. A drawback was that it required custom architectures for each task this was more of a building block. The works of Abkib et al. for Flair [1] follow this direction.

Howard and Ruder showed with Ulmfit [10] how to finetune from an ordinary language model to downstream tasks that broke multiple SOTAs[10]. They use an existing language model and have several methods and ways for this. They bet several state-of-the-art performances for text classification and sentiment detection. The OpenAI transformer by Radford et al. [28] also starts with an ordinary language model and 'transforms' it to a downstream task. Their performance of text classification are worse than Ulmfit but they managed to solve more tasks. For example, they also did text matching such as question answering. There is a comparison of several approaches done by Perone et al. [24]. Sadly without Ulmfit or the OpenAI transformer.

3 CONTEXT-AWARE CLASSIFICATION OF NEWS COMMENTS

We want to classify news comments by looking at the comments text as well as the article and surrounding comments. This is a combination of text classification and text matching. We follow the recent works of language model finetuning as done by Ulmfit [10] or the OpenAi Transformer [28]. We will first present several baselines before describing the model in detail.

3.1 Baseline

First, we classify the news comments without relation to the article or other comments. We will use traditional feature-based approaches as described by [13]. Average word length, average sentence length, and comment length were a good predictor. But also use stylometric features as used by Potthast et al. [26]. We will as well use Umfit [10] to classify comments. Then we also want to relate the comment to the article with some simple distance measures such as cosine distance.

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

[9] https://en.wikipedia.org/wiki/Language_model

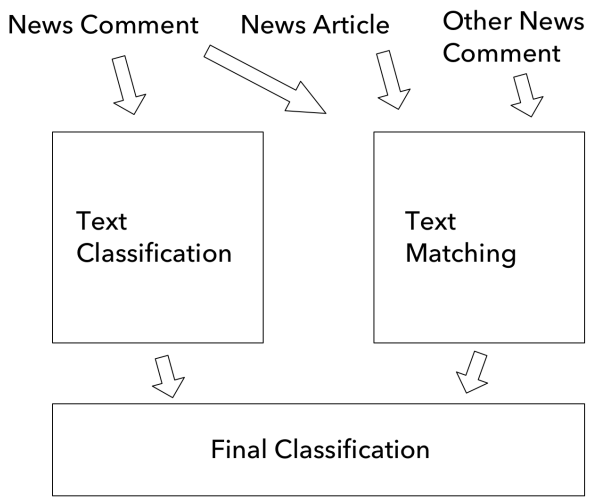

Figure 1: Overview of the model architecture.

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 comments text. Both sub-structures are combined into a final layer to classify the comments. For the text classification, we follow the work of Ulmfit [10]. For the text matching, we follow the work of the AI transformer [28]. 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 for our task. An overview of the complete module is shown in Figure 1. For our use case, this 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

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.

[10] http://nlpprogress.com/text_classification.html

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) [14] [11], in English	10k articles with their 663k comments from 303k comment threads, from 2012 to 2016, from Canadian newspapers	1,043 annotated comments in responses to 10 articles, labeled for constructiveness and toxicity
Yahoo Annotated News Corpus (YNACC) [12][19], in English	522k comments from 140k threads posted in response to Yahoo News articles	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) [31, 32] [13], in German	12k articles, 1M comments, from an Austrian Newspaper	11k comments with the following labels: sentiment, off-topic, inappropiate, discriminating, feedback, personal studies, argument used
Tencent News Corpus (TNC) by Qin et al. [27], in Chinese	200K articles and 4.5M comments, from a Chinese news website, 2017	40k comments labeled for quality (from 1 to 5)

For this work, the YNACC for English and the OMPC for German will be used 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.

4.2 Turning User Votes into Classes

Although there exists some labeled data, the amount is still relatively small. Halevy et al. [8] 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 up-votes. Because user votes are common in news sections, and there is an abundance of news comments available, 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 upvotes for each comment
(7) Classify the comments with the most upvotes as positive and the least as negative, leaving out the middle

One idea is that do not take the upvotes 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 upvotes 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

5.1 Machine Learning

To performance of the machine learning method described in Section 3 will be evaluated on the datasets in Table 4.1.

5.2 Turning User Votes into Classes

To evaluate our method of normalizing the up-votes proposed in Section 4.2, we compare the resulting classification against the golden truth of the comments in Table 4.1. There are certain labels such as 'constructiveness' that should occur significantly more often in 'popular' comments. In a second step, we conduct a user study (N^{14}=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. 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] 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

[4] 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.

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

[6] 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

[7] 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

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

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

[10] 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.

[11] 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).

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

[13] 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

[14] Varada Kolhatkar, Hanhan Wu, Luca Cavasso, Emilie Francis, Kavan Shukla, and Maite Taboada. 2018. The SFU Opinion and Comments Corpus: A Corpus for the Analysis of Online News Comments. (2018).

[15] 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

[16] 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

[17] 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

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

[19] 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.

[20] 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

[21] 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.

[22] 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

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

[14] number of participants

[24] 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

[25] 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).

[26] 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

[27] 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

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

[29] 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.

[30] 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

[31] 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.

[32] 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

[33] 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

[34] 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

[35] 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.

[36] 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