Sentiment Analysis on News Articles and Their Comments

Course project for CSE 6240: Web Search and Text Mining, Spring 2020

Mohit Joshi
Georgia Institute of Technology
Atlanta, Georgia, USA

Ruslina Utomo
Georgia Institute of Technology
Atlanta, GA, USA

Sahil Dhingra
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

In this paper, we outline our methodology for exploring the relationship between news articles and their comments. We use machine learning models to look for certain sentiments found in comments and then apply this learned model towards the corresponding articles to predict its sentiment. For our baselines, we conducted 2 separate yet related experiments derived from the University of Western Ontario [5], one experiment testing the 5 class case of sentiment classification and the other testing the 3 class case. For our own methods, we created two machine learning models, one being a recurrent neural network based model and the second being a Bidirectional Encoder Representations from Transformers model. While our proposed methods surpassed both of our baseline measurements, we did not find the strong relationship we hoped to find between an article's sentiments and its comment's sentiments. However, we believe that this methodology could be extended towards a study that uses the same type of text between source text (article) and the response text (comment).

CCS CONCEPTS

• **Applied computing** → *Law, social and behavioral sciences*;

KEYWORDS

datasets, neural networks, sentiment analysis, text mining

ACM Reference Format:
Mohit Joshi, Ruslina Utomo, and Sahil Dhingra. 2020. Sentiment Analysis on News Articles and Their Comments Course project for CSE 6240: Web Search and Text Mining, Spring 2020. In *Proceedings of* . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Our objective for this paper was to explore the sentiment relationship between articles and their corresponding reader comments. Does a particular type of article elicit a certain type of response? Finding a relationship between the two could be useful information for both the reader and the writer. A writer can scan their written work for possible unwanted reactions prior to publishing and a reader could have advanced sorting and filtering tools based on this relationship.

In order to explore our objective, we needed an annotated comments dataset and an annotated articles dataset. We procured an annotated comments dataset through the Yahoo NLP Dataset catalog and then we crawled the web to gather the corresponding articles to the comments. Since the articles were obtained as raw texts from the internet, there were no sentiment labels that came with it. In order to relate the articles to the annotated comments, we hand labeled the articles to match the sentiments aggregated from the comments datasets. There were far more labeled comments than articles and both of our methodologies relied on training a model through the labeled comments first and then applying this model to the articles to predict the article sentiments. The sentiment prediction would then be compared to the ground truth labels and should the predictions be substantially accurate, it would imply that a relationship exists, and then we could delve into the relationship.

Our first method revolved around an RNN model, to take advantage of the fact that there is an inherent order to our sentence input data, and a pre-trained Word2Vec model to make use of the sheer amount of information already available to us on the web. Our second method involved a Bidirectional Encoder Representations from Transformers [2] model to benefit from the built-in attention mechanism and positional encoding. Both models were trained using the comments dataset and were able to exceed the comment sentiment prediction accuracy scores from our baseline experiments. However, when we applied the models toward the articles, the prediction scores were extremely low and implied that there was a lack of relationship between the article and comment sentiments. We believe that perhaps the difference in writing style, language, and grammar between the comments and articles is confusing for the model. In addition, articles inherently have weaker sentiments compared to individual comments. Perhaps instead of focusing on articles and comments, to mitigate the difference in

Unpublished working draft. Not for distribution.

style between the two, a new similar study could be applied to text of the same type, such as tweets and its responses. Or instead of focusing on predicting sentiments, a study could be performed to generate a distribution of response sentiment probabilities. Although our study did not net the relationship we had hoped for, it does pave the way towards a different focus for future work.

2 LITERATURE SURVEY

2.1 N. Diakopoulos and M. Naaman; Topicality, Time, and Sentiment in Online News Comments[3]

This paper examines the relationships between news comment topicality, temporality, sentiment, and quality. It is found that particular topics evoke more negative comments and could be labeled for additional moderation attention. From temporal analysis of comments, it is observed that frequency of a user commenting while correlated to the negativity of their comments, is also connected to their propensity to be blocked by the moderators.

2.2 M. Burke and R. Kraut; Mind Your Ps and Qs: The Impact of Politeness and Rudeness in Online Communities[1]

This article studies the effect of politeness on reply rates in online discussion forums. Politeness is measured by combining two approaches - driven by theory and bottom-up perception using surveys - to train a machine learning model. The authors find a large difference in politeness across groups and its effect on responsiveness - politeness eliciting more responses in technical groups while rudeness being more effective in political groups.

2.3 N. Diakopoulos and M. Naaman; Towards Quality Discourse in Online News Comments[4]

This paper examines the needs and desires of commenters and how different journalistic approaches work towards managing comment quality. The users motivations for reading or writing comments can be geared towards seeking gratifications in different forms, like information, personal identity, integration and social interaction, and entertainment. Providing users with advanced tools to filter or sort comments, based on metrics such as sentiment or relevance, could help alleviate users expectations and therefore the overall quality of the comments.

2.4 Shortcomings

- Relationship between the content of a particular article and the tonality and informational content in the comments has not been studied, and any causality or contextual linkages at an article level have not been established.
- Since the articles have been grouped into different categories, it is yet to be studied how the distribution of sentiments in comments varies across different articles within a category.
- While using reply counts as a proxy for responsiveness, the quality, emotional and informational content of the responses has not been studied.
- Relevance and usefulness of comments with respect to the article and across different sentiments is still not established.

3 DATASET DESCRIPTION & ANALYSIS

3.1 Data Preparation

Our data consists of The Yahoo News Annotated Comments Corpus (YNACC) [6] and its corresponding news articles. The annotated comments dataset was downloaded from the Yahoo NLP Dataset catalog and the corresponding internet articles were crawled and downloaded from the web.

In order to use the web crawled articles, we processed the mined text by removing the HTML and formatting tags just to extract the content of the text. After the text was processed, we selected articles and hand labeled them for the same sentiment categories outlined in Table 5 and a single reaction field as well. The YNACC dataset was processed to remove exact duplicates and the sentiment labels that will be used for our experiments were extracted into a binary vector.

Our experiments revolve around building a model that will be trained on the sentiment classifications in the annotated comments. The comment dataset already has the sentiment classifications we need to train the model, by hand labeling the articles in a manner that correlates to the labeling of the comments, we have the groundwork to work on finding a relationship between the two datasets.

3.2 Raw Dataset Properties and Features

There are two types of datasets that will be used. The first are the comments dataset downloaded from Yahoo's NLP Dataset catalog and the second are the corresponding articles to the comments. From Yahoo's NLP comments dataset, there were three sets of annotated comments that were provided:

- Expertly Annotated Comments (EAC Dataset)
- IAC Annotated Comments (IAC Dataset)
- Turk Annotated Comments (TAC Dataset)

Each dataset is made up of comments taken from the response section of an internet article. The topic range for the articles is broad. Comments that are direct responses to the article and not a response to another comment is classified as the start of a subdialogue (SD). Any comment that is a reply to a comment will be classified in the same subdialogue as the parent comment. There are multiple subdialogues per URL listed as an article of note would hopefully have more than one comment in response. Each comment is annotated for sentiment characteristics, but while there are duplicate comments in the datasets, only approximately 10% of the comments are true duplicates. For the remaining duplicates, although the comment text and identification is the same, the sentiment characteristics were different as a single comment can have multiple sentiment characteristics assigned. Table 1 displays the number of unique comments, subdialogues and articles versus the total number of comments for each dataset.

From the three datasets the raw word count statistics are described in Table 2. Out of the unique articles identified, 1,350 articles were crawled and downloaded and the word count statistics are described in Table 3. The comments are markedly shorter than the

Unpublished working draft. Not for distribution.

Data Set	Total Comments	Unique Comments	Unique Subdialogues	Unique Articles
EAC	23,383	9,160	1,400	680
IAC	22,536	7,419	986	999
TAC	25,532	8,540	1,300	536

Table 1: Count of total comments, unique comments, unique subdialogues and unique articles

articles in length and tokens per comment, but due to the shear amount of comments, the vocabulary size is much greater in the comments dataset than the articles dataset. A comparison of the two datasets can be seen in Table 4.

Word Count	EAC Dataset	IAC Dataset	TAC Dataset
Min	0	0	0
Mean	46.21	89.01	44.96
Median	25	44	25
Max	2,433	4,097	2,481

Table 2: Word count statistics of the three different datasets

Downloaded Articles	Min	Mean	Median	Max
1350	0	418.72	242	23,934

Table 3: Word count statistics of downloaded articles

Statistics	Combined Comments	Articles
Average # of Sentences	3.5	20.2
Average # of Tokens	28.55	329.40
Vocabulary Size	6110	1612

Table 4: Statistical comparison between the articles and all three datasets combined

3.3 Data Analysis

The EAC, IAC, and TAC dataset have similar sentiment annotations but there are slight idiosyncratic differences. The EAC dataset had the most amount of detail and sentiment break down compared to the IAC and TAC datasets. The IAC and TAC datasets had the exact same data fields present, the EAC dataset had all data fields present in the IAC and TAC as well as additional fields that further detailed the sentiment characteristics. Table 5 displays the labels that were given in each dataset and how certain labels will be grouped together. Figure 1 shows the numerical breakdown of how the comments fall into the five sentiment classification outlined in Table 5.

Shared fields in EAC, IAC, TAC:
SDID (SD: SubDialogue), CommentIndex, Headline, URL, GUID, CommentID, TimeStamp, Thumbs-Up, Thumbs-Down, Text, ParentID, ConstructiveClass, SD_Agreement, SD_Type

Fields Only in EAC:
Sentiment, Tone, CommentAgreement, Topic, IntendedAudience, Persuasiveness

Sentiment Classification	EAC	IAC	TAC
Humor	Snarky/ Humorous	Funny	Snarky/ Humorous, Funny
Sarcastic	Sarcastic*	Sarcastic	Sarcastic
Positive	Positive/ Respectful, Positive**	Warm/ Sympathetic	Warm/ Sympathetic, Positive/ Respectful
Controversial	Contro- versial*	Contro- versial	Contro- versial
Argumentative	Argumen- tative, Flamewar	Argumen- tative	Argumen- tative, Flamewar

Table 5: Sentiment classification from SD_Type field
***Sentiment classification from Expert/Tone field**
****Sentiment classification from Expert/Sentiment field**

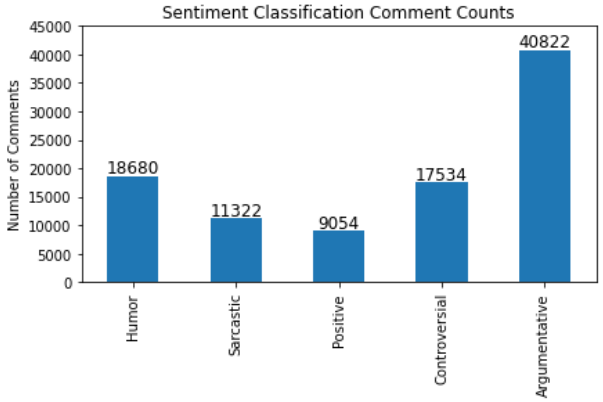

Figure 1: Numerical breakdown of sentiment counts that fall within the 5 classifications described in Table 5

In the three datasets, the majority of the sentiment data that will be used is stored in the SD_Type field. In the EAC dataset, the Sentiment field and the Tone field will also be used to supplement the SD_Type field to match the sentiment types provided by the IAC and the TAC dataset.

In the EAC dataset, it is interesting to note that while we have grouped the Positive label in the Sentiment field and the Positive/Respectful label in the SD_Type field we did not group the

Unpublished working draft. Not for distribution.

Negative label in the Sentiment field with the Argumentative, and Flamewar SD_Types, and the Controversial label in the Tone field due to the low correlation between. Table 6 depicts the Pearson Correlation Coefficient between these fields along with an anti-correlation between positive and negative labels.

	EAC Sentiment: Positive	EAC Sentiment: Negative
Positive/Respectful	0.59	-0.20
Argumentative, Flamewar, Controversial	-0.31	0.17

Table 6: Pearson Correlation Coefficient between EAC Sentiment types and the EAC SD_Types and Tone types.

To relate the articles that were downloaded to the sentiments that were found in the annotated comments, we hand labeled 260 articles to have matching sentiment categories. The breakdown of the manual analysis of the article sentiments is shown in Table 7.

Humor	Sarcasm	Positive	Controv-ersial	Argumen-tative
59	41	86	159	90

Table 7: Count of sentiment classifications found in hand labeled articles

In addition to the ground truth labels that came with the annotated datasets, we also used a Word2Vec model to create word clusters to capture the similarity in contexts between the words. After several studying several iterations in our 3 Classes experiment, we determined that 20 clusters gave us the best results. The cluster averages that we found most effective were:

[1.17, 0.74, 0.44, 0.50, 1.57, 1.22, 0.90, 0.41, 3.36, 0.58, 0.29, 0.42, 0.67, 0.76, 0.73, 0.77, 0.63, 2.07, 0.12, 0.52]

4 EXPERIMENTAL SETTINGS AND BASELINES

In this section, we describe the two baselines - the first baseline from a published paper and the second baseline created by us, for which we discuss the experimental settings as well.

4.1 Baseline Description - Baseline 1

Both of our baselines were derived from a study from the University of Western Ontario titled "The Use of Text Similarity and Sentiment Analysis to Examine Rationales in the Large-Scale Online Deliberations" [5]. In this study, the researchers attempted to train a sentiment classifier using small snippets of text and then evaluate that classifier on larger strings. Specifically, they manually labeled 961 two-word phrases with one of 3 classes (positive, negative, and neutral), and used them to train a machine learning model (KNN in their case specifically). This is useful to us because it closely mirrors the fact that we also plan on training a model using small strings (comments) and then evaluating the model on larger string (articles). Once the model was trained, they again manually labeled 236 sentences from the Wikipedia AfD forum and used these as their evaluation metric. They received an accuracy of 45.7 percent, which is what we will use as our first baseline. We believe that this study serves as a good baseline for our experiment for a few reasons. The first is that this experiment was very close to ours in terms of theme and scope; both have to do with sentiment classification on text from public forums. This is useful because this allows us to more closely and accurately compare the results from both experiments. The second reason is that this experiment trained and evaluated its machine learning model on similar data to what we planned on using. Specifically, they trained their model on small snippets of text (2 word phrases) and evaluated the model on longer strings (full sentences), which we trained our model on short comments and evaluated it on the full article text. Our second baseline will explore the accuracy of a 5 sentiment classification model and a 3 sentiment classification model.

4.2 Experimental settings - Baseline 2

In this study, we set up two separate yet related experiments in order to test the validity of our hypothesis, one for each baseline.

In the first experiment, we first created a Word2Vec model which was trained using the text from all 66,000+ comments. We used this model to create vector representations of each comment, and then feature vectors for each comment were constructed by averaging the word vectors for all tokens within that comment. These feature vectors were then paired with their corresponding sentiment labels and used to train a neural network to detect those same sentiments. In the case of multiple sentiments, the model will attempt to predict all labels, so a sigmoid activation was used instead of a softmax. In this experiment, there was a 80:20 train/test split used for the training of the neural network, which allowed us to perform 5-fold cross validation. The network itself had 2 fully connected hidden layers with 256 neurons in each layer, the optimizer used was Adam, and the learning rate was 1e-4. The loss function used for the model was categorical cross-entropy, since we had 5 categories which represented our sentiment classes, and the evaluation metric was accuracy. The neural network was trained on a Google Colab instance with 68 gb of GPU space allocated.

In the second experiment, we used a different 3-class sentiment value as the target, with sentiment values positive, negative and mixed, but this is available only for the Expert dataset. The dataset of 23,000+ was split into train/test in 80:20 proportion. By training a Word2Vec model on the comments dataset, the words in the dataset were converted into Word2Vec vectors and then grouped into 20 KMeans clusters. Each cleaned comment is then converted into a bag of centroids feature vector, where each vector component represents the count of the number of words in that comment belonging to a particular cluster. Using this feature set, best hyper-parameters (number of trees) were selected using 5-fold cross validation and used to train a Random Forest classification model, with gini as the splitting criterion and max depth as none. Similar to the first experiment, this was trained on a Google Colab instance.

Unpublished working draft. Not for distribution.

5 PROPOSED METHOD

To build upon previous work and develop a model with better accuracy for sentiment prediction, we move towards the latest and state-of-the-art deep learning methods, such as RNNs, LSTMs, Transformers. Instead of generating word embeddings and using an average embedding or cluster frequency as features in a different ML model for prediction, these methods encode the contextual information from the training data and preserve the positional information for prediction.

5.1 RNN based model

In our baseline experiments, we used simple machine learning models to attempt to capture the relationship between sentiments. In order to build upon that, we decided to explore the use of Recurrent Neural Networks (RNNs) in our first proposed method. The individual cells within an RNN network are more complex in the fact that they each keep a hidden state that is updated using the input data as well as the previous hidden state. This means that RNN networks can use the order of the input as useful information, and take advantage of the fact that the input data may have an inherent order to it. Since we are working with word tokens within sentences, it is very much true that the order of the input does carry important information.

The second major change we made for this proposed method was moving from a self-trained Word2Vec model to one that has been pre-trained. In our baseline experiment, we trained a Word2Vec model using the tokens found in our original data, however this dataset was relatively small and likely did not contain a good representation of the typical word distribution of online discourse. To mitigate this, we started using the Global Vectors for Word Representation (GloVe) [7] model, which has been trained on over 2 billion tweets and 27 billion tokens. We believe that the massive increase in training set size will give much better vector representations in our experiments and hopefully will result in an increase in accuracy.

Our proposed method start with downloading the Twitter version of the GloVe model and taking an intersection of its vocabulary with the vocabulary we obtained from our original dataset. Using the word vectors from the resulting model, we can construct feature vectors for each comment by averaging together all of the word vectors for the tokens within that comment. Then, we trained an RNN model (SimpleRNN from the Keras library) using the following hyperparameters:

- Layers: 2
- RNN units/layer: 256
- Hidden state size: 7
- Learning rate: 1e-4
- Epochs: 1.0
- Train test split: 80:20

5.2 BERT Transformer model

A more sophisticated architecture than recurrent networks is that of transformer models, which have recently proven to be more powerful and better in terms of accuracy. For our experiments, we use a pre-trained transformer model developed by Google in 2018, BERT - Bidirectional Encoder Representations from Transformers [2].

A transformer model consists of two main components - a set of encoders chained together and a set of decoders chained together. The encodings generated contain information about the parts of the inputs which are relevant to each other, which is then used by each decoder for processing, using their incorporated contextual information. This is achieved using attention mechanism in both encoders and decoders, which for each input, weighs the relevance of every input and draws information from them accordingly when producing the output. BERT uses multi-head attention that further improves The performance, by implementing several attention mechanisms. By this mechanism, transformers are able to outperform other methods, since they are able to preserve contextual information in a more robust manner. This model performs much better than both the baselines, which are more simpler in the predictive aspect as they don't involve a recurrent model, but only feeding aggregated embeddings into an ML classifier. While Word2Vec embeddings do incorporate context, using these embeddings in an aggregated manner, like frequency of clusters of embeddings in our baseline, loses contextual and temporal information for a word in a particular sequence. Unlike Word2Vec, Transformer models would generate different embeddings for the same word depending on its position and hence have a different effect on the output for different sequences, thus incorporating the important aspect of the order of words in the sequence.

Our BERT model was tuned on the EAC dataset for 3-class sentiments (positive, negative, mixed). Due to the limit on maximum length of an input sequence for BERT, comments and articles longer than 512 tokens were split and used as a separate record, however the number of such records was negligible. For training procedure to tune the BERT model, TensorFlow library was used. BERT library was used to import the pre-trained model as well as for tokenizing the input records and the optimization module. A single layer - with Dropout, Matmul, Bias addition and Softmax modules - was added to BERT to adapt it to our training task. Hyperparameter values used for training can be summarized as follows:

- Batch size: 16
- Learning rate: 2e-5
- Epochs: 2.0
- Warm-up proportion = 0.1
- Train test split: 80:20

6 EXPERIMENTS

The baseline results for experiment 1 (5-class sentiments) and experiment 2 (3-class sentiments) can be seen in Figure 2.

For the first experiment, with 5-sentiment classification, an accuracy of 38.2 percent was achieved.

For the second experiment, with 3-sentiment classification, an accuracy of 57.2 percent was achieved. The accuracy is better for negative and mixed sentiments and lower for positive sentiment. The accuracy is not much lower than the baseline from the published paper. However, it seems to be better for two of the classes and worse for the third. Sampling or class balancing might be able to improve the accuracy for this experiment.

Unpublished working draft. Not for distribution.

In terms of performance of our proposed methods, our first method using RNNs on 5-class sentiments (43.7 percent accuracy), outperforms baseline created by us (38.2 percent accuracy) but has a slightly lower accuracy compared to the paper's baseline (45.7 percent accuracy), on the comments dataset. The second method using BERT on 3-class sentiments (65.2 percent accuracy), outperforms both the baselines (60.2 percent accuracy for the paper and 57.2 percent accuracy for our baseline). A performance summary of baselines and proposed methods can be seen in Figure 2.

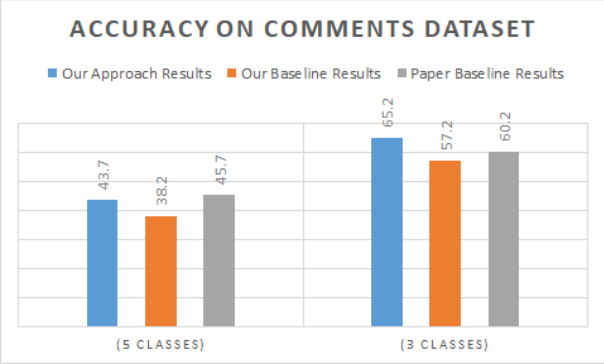

Figure 2: Final results for Baseline and Proposed Methods - Accuracy for each sentiment class on comments dataset 5

As expected, deep learning methods (RNNs and BERT) are able to outperform the simpler methods used in our baselines, and even the original paper in the case of BERT. This highlights the improvement in accuracy by incorporating order and context into prediction for a sequence, compared to an aggregating approach where similar words are aggregated and the concept of order is not clearly incorporated. Even with a better accuracy on comments, the accuracy of sentiment prediction on corresponding articles is fairly low - slightly better than random chance for RNN (5 class) and 46 percent for BERT (3 class). Due to the low accuracy on predictions for articles, the model can't reliably be used to understand the sentiment of an article and therefore a relationship between an article and its sentiments using the textual information cannot reasonably be understood. In the next section, we postulate possible reasons behind this result.

7 CONCLUSION

The accuracies we obtained from evaluating our models on the text from the articles were all very low, not much better than random chance. Some possible explanations are:

- There is a weak/no relation between comment and article sentiment
- The relation between comment and article sentiment is too complex for our models
- The difference in writing style, language, and grammar between comments and articles is confusing the models
- By nature, articles are neutral or have weaker sentiments compared to individual comments

To address the drawbacks of our approach, there are several possible ways that we could develop a better classifier for predicting sentiments for articles:

- In order to mitigate the differences in style between the comment and article text, we could pull all data from the same website (e.g. using a tweet and its comments)
- Instead of focusing on sentiment, we could explore the relationship between the stance of an article and the stance of their comments (i.e. agree, disagree, neutral)
- Instead of predicting a single sentiment for an article, we could generate a distribution of sentiment probabilities

8 CONTRIBUTION

All team members have contributed a similar amount of effort.

REFERENCES

[1] Moira Burke and Robert Kraut. 2008. Mind Your Ps and Qs: The Impact of Politeness and Rudeness in Online Communities. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work* (San Diego, CA, USA) *(CSCW '08)*. Association for Computing Machinery, New York, NY, USA, 281–284. https://doi.org/10.1145/1460563.1460609

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[3] Nicholas Diakopoulos and Mor Naaman. 2011. Topicality, Time, and Sentiment in Online News Comments. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 1405–1410. https://doi.org/10.1145/1979742.1979782

[4] 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* (Hangzhou, China) *(CSCW '11)*. Association for Computing Machinery, New York, NY, USA, 133–142. https://doi.org/10.1145/1958824.1958844

[5] Wanting Mao, Lu Xiao, and Robert Mercer. 2014. The Use of Text Similarity and Sentiment Analysis to Examine Rationales in the Large-Scale Online Deliberations. https://doi.org/10.13140/2.1.4779.9687

[6] Courtney Napoles, Joel Tetreault, Enrica Rosata, Brian Provenzale, and Aasish Pappu. 2017. Finding Good Conversations Online: The Yahoo News Annotated Comments Corpus. In *Proceedings of The 11th Linguistic Annotation Workshop*. Association for Computational Linguistics, Valencia, Spain, 13–23.

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

Unpublished working draft. Not for distribution.