Classification of Deliberation in Facebook Comments Using Machine Learning

Anders Schill
D13, Lund Institute of Technology
dat13asc@student.lu.se

Oscar Svensson
D13, Lund Institute of Technology
elt11osv@student.lu.se

Abstract

This paper describes the usage of neural networks in an attempt to classify political Facebook comments as deliberative or non-deliberative. Different preprocessing techniques are explored and evaluated in search for an optimal combination. The study did not result in any classifiers that can be used to solve real-world problems. However, some conclusions can be drawn about what preprocessing techniques contribute to a better result.

1 Introduction

The purpose of this project was to explore the possibility of automatically classifying two million user comments, gathered from various Facebook pages, as deliberative or non-deliberative using machine learning, in order to aid in another study (Segesten, 2017) where the correlation between deliberation and other parameters was of interest.

All comments were previously unlabeled and gathered from Facebook pages belonging to different Brexit campaigns. 3000 comments were manually labeled by four different people and used to train neural networks using a bag-of-words model. The three different classifiers were then combined to achieve the end goal.

Some success was had in classifying the comments and certain parameters used in the preprocessing of the text were shown to improve the result. However, the results are not nearly good enough to use for the intended purpose at this point and a bigger dataset of labeled comments are most likely needed to reach any useful outcome.

2 Background

In order to classify comments as deliberative or non-deliberative. One must first define deliberation. The definition used in this project was based on (Segesten, 2017) and is described below.

A comment is considered deliberative if it can be classified as positive according to three independent requirements: openness, political relevance and respectfulness.

1. Respectfulness
   If the comment contains curse words or negative stereotypes, it is not respectful.

2. Political relevance
   A comment is politically relevant if it contains claims about an issue of political relevance or has references to back up a claim.

3. Openness
   A comment is open if it has an open-ended question, mentions other users or references other groups.

A deliberative comment should, to sum up, be open to the input of others, be written in a respectful and civilized manner and also be justified with an argument, such as a reference to back a claim that is made.

Comments that fulfill all three requirements are rare; in our subsample, around 7% qualified as deliberative.

Here follows an example to highlight this point.

Andree Gillette - 'mendacious' - I like that! ? Just need to now figure out where and when to use

it ?

This comment is defined as open and respectful but it has no argument. Therefore it is not labeled as deliberative even though it is very close.

Unfortunately, there is no easy way to decide whether a comment should be considered deliberative or not and therefore, naturally, there will always be a certain amount of disagreement between annotators. A way to measure this is to calculate the so called inter annotator agreement, which is discussed further in section 7.

3 Description of Data

The dataset used in this project consists of comments collected from three different Facebook pages: Strongerin, LeaveEU and VoteLeave.

In total, the dataset contains 2 317 105 comments gathered from these three sources without any processing.

The lengths of the comments vary from one word to comments that stretch over several lines. The average number of characters is 201 and the median is 98. The distribution is shown in the histogram in figure 1.

Even though the lengths of the comments vary greatly, they are almost exclusively written in English.

A random subsample of 3000 comments was extracted and labeled manually. Almost all comments in the sample were labeled as politically relevant or politically irrelevant. Almost 2000 comments were labeled as open or not open. Almost 2000 comments were labeled as respectful or disrespectful.

Table 1 in Appendix A contains statistics describing the random sample.

4 Methodology

4.1 Preprocessing

Before classifying the comments, some preprocessing can be beneficial. The following preprocessing techniques were used in this project: stemming, removal of stopwords, removal of punctuation characters and linebreaks and the replacement of words with a class.

Replacing words with a class refers to when a certain type of word is replaced with another word that is the same for all words of a specific kind. In this project, all links, large numbers and small numbers are replaced by a tag indicating their existence. Since words within these classes essentially have the same meaning, this is a good way to reduce the number of unique words.

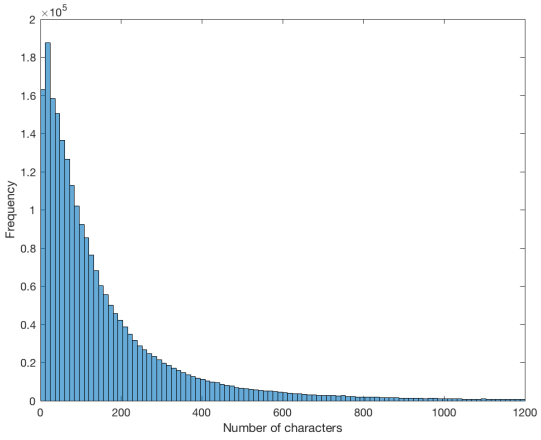

Figure 1: Histogram showing the distribution of comment lengths in the random sample.

4.2 Classification

To begin with, three different classifiers were constructed; one for each rule that defines deliberation. The results were then combined according to the rule that a comment has to be classified as positive by all classifiers to be considered deliberative.

The classifiers were implemented as neural networks using Tensorflow (Abadi et al., 2015) - an open source library developed by Google.

Two different approaches were tried when feeding the data into the neural network. The first method uses words as features. The second one uses so called trigrams. To turn a string into trigrams, every consecutive subsequence of length three is found.

A bag-of-words model was used. By providing features and targets, Tensorflow one-hot encodes the targets. Then, using the bag-of-words encoder, the features are encoded into an embedded matrix. The encoded features are then fed into a fully connected layer with 15 hidden neurons. Lastly, a softmax cross entropy function is used for the output. To optimize the network, the network is added into an Adam optimizer with learning rate 0.01.

5 Results

The parameters used during the preprocessing phase, described in section 4.1, were used in all possible combinations. 5-fold, cross validation was used to evaluate the result. In table 2 in Appendix A, the results of all combinations are presented. The evaluation measures used are micro-averaged f1-score, macro-averaged f1-score, and a confusion matrix.

Since the datasets are unbalanced, the macro-averaged f1-score is preferred over accuracy and micro-averaged f1-score. This becomes clear when considering the case of a classifier classifying all tuples as negative and achieving an accuracy of 0.80 for the political relevance category.

The following combinations resulted in the best macro-averaged F1-scores for each category.

- Respectfulness:
  stemming, remove punctuation, remove stopwords, trigrams
- Openness:
  stemming
- Political relevance:
  stemming, trigrams

To achieve the end goal of classifying the comments as deliberative or non-deliberative, the three classifiers have to be combined. According to the definition of deliberation described in section 2, a comment has to fulfill all three factors in order to qualify as deliberative. Below are the results from a classifier that combines the result from the three individual classifiers.

Accuracy: 0.93
F1-score (micro): 0.93
F1-score (macro): 0.48
Confusion matrix: $\begin{pmatrix} 897 & 0 \\ 71 & 0 \end{pmatrix}$

Another way of achieving this goal is to train a single classifier directly on tuples labeled as deliberative or non-deliberative. The result from such an approach is shown below.

Accuracy: 0.90
F1-score (micro): 0.90
F1-score (macro): 0.50
Confusion matrix: $\begin{pmatrix} 875 & 25 \\ 68 & 3 \end{pmatrix}$

To establish a baseline to which the machine learning methods can be compared, a classifier that classifies a piece of text based on whether any of its words are present in a predefined dictionary, was constructed. For the respectfulness category, the dictionary was constructed from (Dubs, 2011) combined with some common political insults and it produced the following result:

F1-score (micro): 0.73
F1-score (macro): 0.54
Accuracy: 0.73
Confusion matrix: $\begin{pmatrix} 86 & 464 \\ 51 & 1297 \end{pmatrix}$

A method such as this will catch the most obvious disrespectful comments but will fail to detect misspelled words, different word variations as well as other, more subtle, cues like irony or sarcasm.

6 Related work

In a project by Yahoo (Nobata et al., 2016), the possibility of identifying abusive comments in two different domains: finance and news, with the help of a dataset consisting of around 800 thousand and 1.4 million labeled tuples respectively, was investigated. Various different features and their contribution to a successful classification were tried. The features used were based on n-grams, linguistic features and syntactic features. The results from this report shows that a dataset of that magnitude was not necessary to achieve good results and that n-grams contributed the most to a good classification even though a combination of the features was slightly better. The F-scores obtained were 0.795 and 0.817 for finance and news respectively. The report also showed that using labels where all (three) annotators agreed produced a slightly better result.

7 Limitations

Looking at (Nobata et al., 2016), which is similar to the respectfulness classifier, an f1-score around 80% was obtained. The (Nobata et al., 2016) had access to a lot more data and combined different types of

features, which are described in section 6.

More labeled data is most likely something that could improve our results. Using different types of features could also have a positive impact. (Nobata et al., 2016) mentions that the context is often important when classifying a text, and by looking at a whole conversation thread rather than individual comments, a better classification could be made. This would be highly relevant, especially for the openness classifier.

To obtain more labeled data, crowd sourcing would probably be the best alternative since existing datasets would be less useful unless they share the same domain; political insults are for example often different from regular disrespectful language.

The inter annotator agreement between two annotators varied between categories but was found to be in the range 78-82%. This indicates that the labeling of the categories in this study is highly subjective. It was found in (Nobata et al., 2016) that having more annotators label the same tuples improved results.

8 Conclusions

The best macro-averaged f1-scores acquired were 0.59, 0.60 and 0.65 for the categories respectfulness, openness and political relevance respectively. With such low f1-scores the classifiers would not be particularly useful. This becomes even more clear when looking at the best combined classifier which has an f1-score of 0.50.

Compared to the dictionary approach, the machine learning method performs slightly better. A very simple addition to the respectfulness classifier would be to combine it with the dictionary approach either as extra features or in an ensemble.

Since the end goal was to identify deliberation, the combined classifiers are of most interest. One classifier was trained to directly classify comments as deliberative or non deliberative and another used three different classifiers; one for each requirement that defines deliberation. A drawback of the latter approach is that errors could propagate when three classifiers are combined - this is probably the reason that the former performed slightly better.

The combination of preprocessing methods that worked best was different for every category. However, stemming improved the results across the board.

In this study, two different types of features were tried separately: trigrams and words. Theoretically, using trigrams would remove some noise from misspelled words, which are common in the data. Looking at the result, in some cases using trigrams instead of words seem to have improved the result, but this is not the case overall.

9 Acknowledgments

The authors would like to thank Anamaria Dutceac Segesten for supervising the project and Pierre Nugues and Jacek Malec for giving valuable advice on the machine learning aspects of the study.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Jamie Dubs. 2011. Googles official list of bad words, July.

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, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Anamaria Dutceac Segesten. 2017. personal communication, apr.

Appendix A

Table 1: Shows statistics about the random sample.

Category	Positive / Negative	Unique words	Unique trigrams / total trigrams	Total number of comments
Open	580 / 1318 (31% positive)	13850 / 77886	7044 / 423074	2023
Political Relevance	564 / 2308 (20% positive)	17743 / 112573	7834 / 608377	2872
Respectful	1407 / 616 (70% positive)	13850 / 77886	7044 / 423074	2023

Table 2: Different combinations of preprocessing.

Stemming	Remove punctuation	Remove stopwords	Use trigrams instead of words	Open	Politically relevant	Respectful
no	no	no	no	Average accuracy: 0.661727610549 Micro averaged f1-score: 0.661727610549 Macro averaged f1-score: 0.598918340925 Confusion matrix: [[1003, 315], [327, 253]]	Average accuracy: 0.763582801007 Micro averaged f1-score: 0.763582801007 Macro averaged f1-score: 0.632587904586 Confusion matrix: [[1953, 355], [324, 240]]	Average accuracy: 0.702327701723 Micro averaged f1-score: 0.702327701723 Macro averaged f1-score: 0.585802434545 Confusion matrix: [[164, 386], [179, 1169]]
no	no	no	yes	Average accuracy: 0.676486759038 Micro averaged f1-score: 0.676486759038 Macro averaged f1-score: 0.550376523204 Confusion matrix: [[1143, 175], [439, 141]]	Average accuracy: 0.801530439412 Micro averaged f1-score: 0.801530439412 Macro averaged f1-score: 0.550382738822 Confusion matrix: [[2222, 86], [484, 80]]	Average accuracy: 0.718142259183 Micro averaged f1-score: 0.718142259183 Macro averaged f1-score: 0.561299434662 Confusion matrix: [[115, 435], [100, 1284]]
no	no	yes	no	Average accuracy: 0.622542163719 Micro averaged f1-score: 0.622542163719 Macro averaged f1-score: 0.557858615485 Confusion matrix: [[940, 356], [351, 226]]	Average accuracy: 0.740848809672 Micro averaged f1-score: 0.740848809672 Macro averaged f1-score: 0.620092492454 Confusion matrix: [[1845, 419], [314, 250]]	Average accuracy: 0.693558773883 Micro averaged f1-score: 0.693558773883 Macro averaged f1-score: 0.587755485407 Confusion matrix: [[176, 367], [207, 1123]]
no	no	yes	yes	Average accuracy: 0.651334061476 Micro averaged f1-score: 0.651334061476 Macro averaged f1-score: 0.547172434886 Confusion matrix: [[1057, 239], [414, 163]]	Average accuracy: 0.786741243924 Micro averaged f1-score: 0.786741243924 Macro averaged f1-score: 0.576637937491 Confusion matrix: [[2106, 158], [445, 119]]	Average accuracy: 0.696194272259 Micro averaged f1-score: 0.696194272259 Macro averaged f1-score: 0.559054381158 Confusion matrix: [[130, 413], [156, 1174]]
no	yes	no	no	Average accuracy: 0.651612903226 Micro averaged f1-score: 0.651612903226 Macro averaged f1-score: 0.585006568113 Confusion matrix: [[977, 303], [345, 235]]	Average accuracy: 0.759574468085 Micro averaged f1-score: 0.759574468085 Macro averaged f1-score: 0.637242690181 Confusion matrix: [[145, 391], [177, 1147]]	Average accuracy: 0.694623655914 Micro averaged f1-score: 0.694623655914 Macro averaged f1-score: 0.569816819845 Confusion matrix: [[1889, 365], [313, 253]]
no	yes	no	yes	Average accuracy: 0.674731182796 Micro averaged f1-score: 0.674731182796 Macro averaged f1-score: 0.552080363018 Confusion matrix: [[1114, 166], [439, 141]]	Average accuracy: 0.797517730496 Micro averaged f1-score: 0.797517730496 Macro averaged f1-score: 0.551921538276 Confusion matrix: [[2168, 86], [485, 81]]	Average accuracy: 0.705376344086 Micro averaged f1-score: 0.705376344086 Macro averaged f1-score: 0.53962444883 Confusion matrix: [[99, 437], [111, 1213]]
no	yes	yes	no	Average accuracy: 0.630133140114 Micro averaged f1-score: 0.630133140114 Macro averaged f1-score: 0.574351466774 Confusion matrix: [[899, 340], [331, 244]]	Average accuracy: 0.727140255009 Micro averaged f1-score: 0.727140255009 Macro averaged f1-score: 0.618723895755 Confusion matrix: [[1729, 450], [299, 267]]	Average accuracy: 0.684726625004 Micro averaged f1-score: 0.684726625004 Macro averaged f1-score: 0.56235198813 Confusion matrix: [[143, 383], [189, 1099]]
no	yes	yes	yes	Average accuracy: 0.648268574706 Micro averaged f1-score: 0.648268574706 Macro averaged f1-score: 0.568925162456 Confusion matrix: [[977, 262], [376, 199]]	Average accuracy: 0.776320582878 Micro averaged f1-score: 0.776320582878 Macro averaged f1-score: 0.567042295094 Confusion matrix: [[2019, 160], [454, 112]]	Average accuracy: 0.700694381548 Micro averaged f1-score: 0.700694381548 Macro averaged f1-score: 0.570946711148 Confusion matrix: [[140, 386], [157, 1131]]

Stemming	Remove punctuation	Remove stopwords	Use trigrams instead of words	Open	Politically relevant	Respectful
yes	no	no	no	Average accuracy: 0.66068049013 Micro averaged f1-score: 0.66068049013 Macro averaged f1-score: 0.599227499159 Confusion matrix: [[998, 320], [324, 256]]	Average accuracy: 0.771588269454 Micro averaged f1-score: 0.771588269454 Macro averaged f1-score: 0.64283976978 Confusion matrix: [[1969, 339], [317, 247]]	Average accuracy: 0.696502231002 Micro averaged f1-score: 0.696502231002 Macro averaged f1-score: 0.589839778384 Confusion matrix: [[177, 373], [203, 1145]]
yes	no	no	yes	Average accuracy: 0.685474312396 Micro averaged f1-score: 0.685474312396 Macro averaged f1-score: 0.543522247484 Confusion matrix: [[1178, 140], [457, 123]]	Average accuracy: 0.778915021293 Micro averaged f1-score: 0.778915021293 Macro averaged f1-score: 0.652582717667 Confusion matrix: [[1983, 325], [310, 254]]	Average accuracy: 0.716542568622 Micro averaged f1-score: 0.716542568622 Macro averaged f1-score: 0.553974248516 Confusion matrix: [[108, 442], [96, 1252]]
yes	no	yes	no	Average accuracy: 0.648690052341 Micro averaged f1-score: 0.648690052341 Macro averaged f1-score: 0.58750943696 Confusion matrix: [[967, 329], [329, 248]]	Average accuracy: 0.746789854169 Micro averaged f1-score: 0.746789854169 Macro averaged f1-score: 0.635224980396 Confusion matrix: [[1953, 355], [324, 240]]	Average accuracy: 0.688240967957 Micro averaged f1-score: 0.688240967957 Macro averaged f1-score: 0.573833145483 Confusion matrix: [[1837, 427], [289, 275]]
yes	no	yes	yes	Average accuracy: 0.653494375807 Micro averaged f1-score: 0.653494375807 Macro averaged f1-score: 0.562274209841 Confusion matrix: [[1039, 257], [392, 185]]	Average accuracy: 0.780425651253 Micro averaged f1-score: 0.780425651253 Macro averaged f1-score: 0.567008825064 Confusion matrix: [[2096, 168], [453, 111]]	Average accuracy: 0.706851161009 Micro averaged f1-score: 0.706851161009 Macro averaged f1-score: 0.581929942474 Confusion matrix: [[150, 393], [156, 1174]]
yes	yes	no	no	Average accuracy: 0.654301075269 Micro averaged f1-score: 0.654301075269 Macro averaged f1-score: 0.597583644798 Confusion matrix: [[957, 323], [320, 260]]	Average accuracy: 0.771985815603 Micro averaged f1-score: 0.771985815603 Macro averaged f1-score: 0.651986420538 Confusion matrix: [[1915, 339], [304, 262]]	Average accuracy: 0.699462365591 Micro averaged f1-score: 0.699462365591 Macro averaged f1-score: 0.586917922201 Confusion matrix: [[167, 369], [190, 1134]]
yes	yes	no	yes	Average accuracy: 0.672580645161 Micro averaged f1-score: 0.672580645161 Macro averaged f1-score: 0.548887526863 Confusion matrix: [[1112, 168], [441, 139]]	Average accuracy: 0.795035460993 Micro averaged f1-score: 0.795035460993 Macro averaged f1-score: 0.54411570257 Confusion matrix: [[2165, 89], [489, 77]]	Average accuracy: 0.705376344086 Micro averaged f1-score: 0.705376344086 Macro averaged f1-score: 0.543680899986 Confusion matrix: [[104, 432], [116, 1208]]
yes	yes	yes	no	Average accuracy: 0.621930380702 Micro averaged f1-score: 0.621930380702 Macro averaged f1-score: 0.567878129674 Confusion matrix: [[883, 356], [330, 245]]	Average accuracy: 0.728233151184 Micro averaged f1-score: 0.728233151184 Macro averaged f1-score: 0.629096813109 Confusion matrix: [[1707, 472], [274, 292]]	Average accuracy: 0.673106300758 Micro averaged f1-score: 0.673106300758 Macro averaged f1-score: 0.553556600516 Confusion matrix: [[142, 384], [209, 1079]]
yes	yes	yes	yes	Average accuracy: 0.65869335507 Micro averaged f1-score: 0.65869335507 Macro averaged f1-score: 0.580958914298 Confusion matrix: [[988, 251], [368, 207]]	Average accuracy: 0.76393442623 Micro averaged f1-score: 0.76393442623 Macro averaged f1-score: 0.561059764698 Confusion matrix: [[1982, 197], [451, 115]]	Average accuracy: 0.70511427105 Micro averaged f1-score: 0.70511427105 Macro averaged f1-score: 0.591880238 Confusion matrix: [[162, 364], [171, 1117]]