Machine Learning Engineer Nanodegree

Capstone Proposal

Olena Babenko

February 28st, 2017

Proposal

Aggressive text detection in news and articles.

Domain Background

Internet gives people freedom of expression - everyone can say whatever he want anonymously. But it also provide freedom to express hate and aggression against others. And it's hard to protect people from this. Of course, there is always an option to switch off computer, but then all the other good things you can find in internet will be lost. The best solution would be to provide people automatic solution, that warn people about content with extremely negative attitude.

It's a problem that sentiment analysis could solve. Sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic [1]. This problem is not a new one. Related research about analysis of aggressive text detection was made by Laura P. Del Bosque and Sara E. Garza [2]. In this work was used not a binary classification but regression approach by finding score of aggression. Mentioned research was focused to identify cyberbullying in social networks. In current work focus will be on more impersonal aggression in news.

Problem Statement

There are a lot of publication, that describes sentiment analysis on messages in Social Networks. Main focus of this work is on classifying news and articles with aggressive or hate intent. Aggressive text is text that intends to offend other people. It usually contains negative words such as 'angry', 'detest', 'awful' and many swear words. SentiWordNet is lexical resource that provide information about word polarity - how positive or negative it is. It will be used to define negative words. Unlike to messages in

Social networks, news could not include personalized words like 'you' and 'I'. Aggressive news classification will be treated as binary classification problem. Main goal will be to label text as 'aggressive' and 'nonaggressive'. In this project semi-supervised learning will be used, because news dataset doesn't contains labels. Suggested solution could be used later to identify hate news online.

Datasets and Inputs

There is Google Chrome extension BSDetector[1], that warn user about web sites that could contain fake or hate news. Extension checks if website is in blacklist of domains. This list maintained manually. Of course, it's not a perfect solution. This detector was used to scrabble data from 244 web sites by Megan Risdal and was published on kaggle platform[2]. This dataset contains 12,999 posts. Data set is publicly accessible and released under CC0: Public Domain License[3].

Originally this dataset was published to detect fake news, but this work is not about fake news detection. This data set was chosen because fake news authors sometimes use aggression to provoke conflicts. Data set include aggressive and nonaggressive news. It's expected, that dataset contains more nonaggressive news, than aggressive. Data set has feature 'type' defined by BSDetector, but it has many errors. Sometimes, news marked as 'hate' are neutral. The only used feature from this dataset is "text", that include text of the story. Semi-supervised approach will be used to get answers from this one feature only.

Solution Statement

Aggressive news detection is binary classification problem. Data set is too big to label all of the items manually. So there were chosen unsupervised learning approach. Solution model should label text as "aggressive" or "nonaggressive". Text itself couldn't be used directly in classifiers. All text should be processed to find new features that will help in classification. There will be features that shows text polarity and objectivity using SentiWordNet[1] and features that shows amount of negative emotions extracted with WNAffect python script[4]. This additional features could be used to train classifier or test unseen data. A solution models will be DBSCAN and K-Means Classifiers from sklearn library[5] to generate clusters. For each generated cluster will be selected 10-20 samples

[1] BSDetector is available at https://github.com/selfagency/bs-detector
[2] Fake news kaggle data set https://www.kaggle.com/mrisdal/fake-news
[3] About CC0: Public Domain License https://creativecommons.org/publicdomain/zero/1.0/
[4] WNAffect is available at https://github.com/clemtoy/WNAffect
[5] Available at http://scikit-learn.org/

to label them manually. After that labels will be spreaded for all clusters with LabelPropagation[5]. Finally, SVM supervised learning model will be trained with generated labels. This final model will classify data unseen before. To test solution models, 200 articles will be selected in random order and labeled manually.

Benchmark Model

This dataset has feature 'type' with predefined label. Simplest approach is to use this column as a benchmark result. 'Aggressive' label is 'hate' type and 'nonaggressive' are any other types. This row will be compared to manually labeled test set and measured with f1-score. This will show score of basic BSDetector implementation.

But this benchmark results are not good enough, that is why I suggest to use also benchmark model based on research made by Jun-Ming Xu, Xiaojin Zhu [3] that is related to the subject. This research describes building bullying-or-not text classifier. There was chosen SVM as a classifier. The best SVM model obtained with RBF kernel, C = 1000 and γ = 0.1. This model will be a benchmark model. It will be trained with same data as solution models and evaluated by the same metrics.

Evaluation Metrics

In this project unsupervised learning will be used. But test set requires labels to evaluate final solution. That is why testing data set will be labeled manually. Test data set will consists of 200 articles. Articles will be chosen and checked randomly. In this case, there is a big chance, that in the test set will have similar to original dataset distribution. If in test set will be less than 10 'aggressive' articles, they could be added to the test set by searching words 'hate', 'aggressive', 'stupid', 'bastard' in text editor.

F1-score will be used as main metric to compare manually defined labels and predicted results. F1-score is chosen because dataset expected to be unbalanced (nonaggressive news should dominate) and F1-score works fine with such datasets.

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$,where $$\text{Recall} = \frac{tp}{tp + fn}$$ $$\text{Precision} = \frac{tp}{tp + fp}$$

Solution could be used to improve online BSClassifier. In this case classification time of one article is also an important metric.

Project Design

This project starts with defining test set. Main problem is that there are more than 12 000 items in this dataset and data is unlabeled. This is why semi-supervised approach will be used. 200 articles will be selected as a test set and labeled manually. This articles will be picked randomly. If by chance, there will be less than 10 'aggressive' texts, testing dataset could be extended manually. If by chance, there will be more than 50 'aggressive' texts, some of them should be excluded from testing dataset.

Second step will be a text preprocessing. Not all words are not actually needed for future processing: numbers or names (words, written in uppercase, but not in the beginning of the sentence) will be excluded from the data set. Also, stop words - words with very little meaning, like 'the', 'a', 'and' will be excluded during feature engineering process.

One of the most important step will a feature engineering - new features should be extracted from the text. Moreover, this features should be connected with aggressive intent. Otherwise, it would be hard to define two clusters, where one of them should include only aggressive texts. That is why first 3 features will use text polarity classification. There could be multiple approaches how to merge all words to one feature. It could be an average aggression per words or just amount of highly negative words. Also, CountVectorizer[6] could be used to build features from words and add information about pairs of words. 2-gram or 3-gram could be applied for this purpose . Stop-words should be removed. Default english stop-words dictionary from sklearn will be applied. Next step will be an extraction of words that are an emotions. There could be different approaches: there could be a feature with total number words, that describe emotions or just amount of words, that mean negative emotions. This words could be found with WNAffect[5] python script.

After that, data could be applied for classification. There should be 3 models to compare: benchmark model (SVM) and 2 solutions models: DBSCAN and K-Means. Each model could generate K clusters. For each cluster 10 samples will be selected and labeled manually. After that LabelPropagation will be used to label rest of the dataset. Finally, for each model will be trained the same SVM classifier with fully labeled dataset. First solution would use only default parameters for each model. Then, each model should be tuned with GridSearchCV[6] from sklearn library. Second solution will use improved classifiers.

Finally, there should be a table with final f-score result for each model. Additional table should be added to show running time of each model. Conclusion should include results, best model and result, if it could be used on practice.

References

[1] Sentiment analysis. Wikipedia. https://en.wikipedia.org/wiki/Sentiment_analysis

[2] Del Bosque, L. & Garza, S. Gelbukh, A.; Espinoza, F. & Galicia-Haro, S. (Eds.) Aggressive Text Detection for Cyberbullying. Human-Inspired Computing and Its Applications, Springer International Publishing, 2014, 8856, pp. 221-232

[3] Jun-Ming Xu, Xiaojin Zhu, Amy Bellmore. Fast learning for sentiment analysis on bullying. WISDOM '12 Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, Beijing, China, Article No. 10

[4] Vinita Nahar, Xue Li, Chaoyi Pang, Yang Zhang. Cyberbullying Detection based on Text-Stream Classification. Proceedings of the 11-th Australasian Data Mining Conference (AusDM'13), Canberra, Australia, CRPIT Volume 146 - Data Mining and Analytics 2013, pp 49-57.