semantic evaluation of abusive tweets

Abstract

This report presents the results and models of the shared task on Identifying and Categorizing Offensive Language in Social Media. We performed all the three sub-tasks within our coursework and evaluated different sentence embeddings in conjunction with various supervised machine learning algorithms and evaluated their performance. The SemEval Task provides us with an annotated dataset containing English tweets. The competition was divided into three sub-tasks. In sub-task A systems were trained to discriminate between offensive and non-offensive tweets, in sub-task B systems were trained to identify the type of offensive content in the twitter posts, and finally, in sub-task C systems were trained to identify the target of offensive posts.

1. Introduction

The locating and identifying of offensive content online has gained a significant importance in recent years. Big tech companies like Facebook and Twitter are heavily invested in finding offensive content on their platforms, especially hate speech and online bullying, which are on the rise in the online world. Our task in this coursework is supervised classification in which we have been provided with training data of about 14000 tweets, which are annotated with offensive or non-offensive content and the main target of this report also revolves around the identification of online offensive language and whether that content is directed to someone specific like a person or a group or even a company.

The remainder of this report is organized as follows: Section 2 states objectives & requirements, and gives data mining problem definition. Section 3 explains data format and content, raises data quality issues and expected results. Section 4 describes how to convert the data format, how to clean and filter the data. Section 5 provides 3 different models with its testing accuracy using two different dataset that are Lemmatized Dataset and Lancaster Dataset. Section 6 shows how to get results for evaluation and the best classifier. Finally, section 7 concludes and suggests directions for future work.

2. Business Understanding

The Semantic Evaluation challenge is an ongoing series of evaluations of computational semantic analysis systems. These, evaluations or tasks, are intended to explore the nature of meaning in language and while meaning is intuitive to humans, transferring those intuitions to computational analysis is difficult but ultimately worthwhile. The combined shared text, images and sounds of millions of people across the globe carries a wealth of tacit knowledge and information that we absorb on a daily basis and that allows us to navigate our lives, make decisions and informs us of what is going on in the collective village of the world.

This wealth of information continues to grow with every passing second because of the users of online platforms who freely feed the systems with all kinds of data and do so under the guise of safety, comfort and security when using the internet. When this veil is lifted or when others use the same resources to target and attack individuals, said individuals may stop using the online platforms and stop feeding information into the machine which is something the big tech companies do not want to happen.

Tech companies like Twitter want to keep people using the internet and the free online tools they provide because of how valuable the information provided by users is to them and so these companies invest in technology that helps to curb the spread of abusive speech and behaviour in user generated content. Presumably to improve the online safety of us all, to improve the user experience of their products and most definitely keep the information machine running and the users glued to their screens.

Apart from information that can be mined from user data, companies are also keen on increasing user engagement as this is something that in our modern world can be leveraged for great financial gain and even, for better or worse, political benefit. Users will only stick with platforms that they believe they are safe to interact and share on. This is why there is a business incentive to research, model and evaluate ways of enabling computational semantic understanding of mean spirited tweets.

3. Data Understanding

The Offensive Language Identification Dataset(OLID) dataset contains 13,240 annotated tweets that are used to identify and categorize social media data into three categories. As mentioned in Zampieri et al. (2019a), Subtask A, Subtask B, and Subtask C correspond to the identification of offensive language, categorization of offense, and its target respectively.

Variable	Data Type	Values
id	Numerical	
tweet	String/Categorical	
Subtask_a	Nominal/ Categorical	OFF/NOT
Subtask_b	Nominal/ Categorical	UNT/TIN/ NULL
Subtask_c	Nominal/ Categorical	IND/GRP/ OTH/ NULL

Table 1: Data Type Training File

Variable	Data Type
id	Numerical
Tweet	String/Categorical

Table 2: Data Type Test File1

Variable	Data Type
id	Numerical
Subtask	Nominal/ Categorical

Table 3: Data Type Test File2

Table 1 highlights the data type in the training set for each variable. Test data is divided into two files with the format shown in Table 2 and Table 3. Thus the data is combined using key id field. User data is anonymized and replaced with @USER in place of tweeter id. The data consist of multiple emojis such as '@USER He is 😭😭😭 he is so precious 💖 .' Also, the data contains the use of symbols, punctuations, and a combination of uppercase and lower case.

3.1 Data Quality

The quality of data is good with no missing values, no duplicate values, and zero irrelevant attributes. Data is encoded in the UTF-8 format.

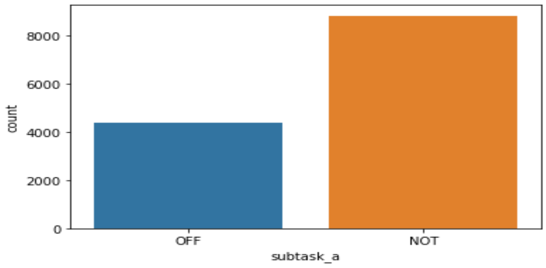

Figure 1. Subtask A.

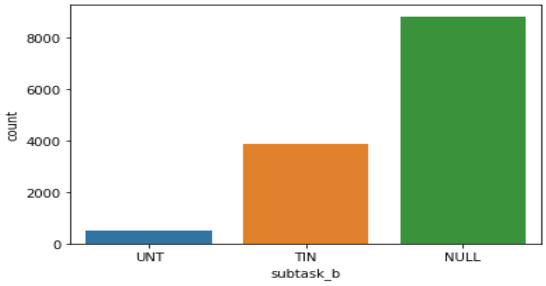

Figure 2. Subtask B

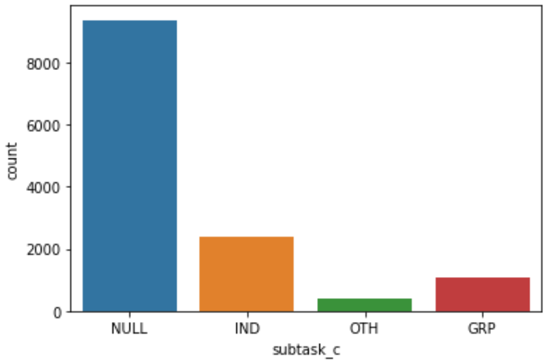

Figure 3. Subtask C

As shown in Figure 1, Figure 2, Figure 3 the training data has an imbalanced classification for subtask_a, subtask_b, and subtask_c respectively.

4. Data Preparation

Twitter data is unstructured data with a collection of emotions, product reviews, opinions from people with a diverse background. Twitter data could be messy consisting of a variety of symbols that include Punctuation, Emojis, Integers, and characters with varying letter cases. Thus a thorough data cleaning process must be established to remove such unproductive variables. Data preparation techniques described in Nikil Prakash (2019) are effective and proposed below.

4.1 Attribute Selection

As part of the Attribute Selection process, in both the methods, the ID field has been removed, since it is a unique field and thus won't contribute much to the train a model. Remaining fields such as string and classes are used to perform sentiment analysis.

4.2 Data Cleaning

4.2.1 Convert all character to lower case

To perform Data Cleaning, in the 1st Method, we have changed the characters to lowercase, removed emojis, punctuations, and Integers while keeping a few parameters such as Hashtag. However, for the 2nd Method, we considered.

4.2.2 Retaining Integers

Integers which could be essential while training the model. Consider below tweet example, wherein the 1st Method, a trending hashtag #Trump2020 was reduced to #Trump. On checking, it was found the hashtag was used more than 20 times. Thus we have considered integers to increase the diversity of vectors.

Method 1: 'home drunk #maga #**trump'**, OFF
Method 2:'home youre drunk #maga **#trump2020** oncoming_fist united_states oncoming_fist', OFF

4.2.3 Retaining Emoji's

Since emoji's convey sentiment, unlike the 1st Method, we have used the emojis package in Python to convert emoji to emoji's code for our 2nd Method. For example, emoji '😊 ' is converted to 'blush'. Similarly, 😟 is changed to 'worried'. Below are tweets derived after using both procedures.

Method 1: 'someon piec shit', OFF
Method2: 'someone piece shit **face_with_tears_of_joy'**,OFF

4.2.4 Applying Stemming and Lemmatization techniques

In order to reduce words to its root for we performed Stemming and used 'PorterStemmer' in the previous method. To improve model accuracy we have considered multiple Stemming and Lemmatization approaches that include 'Lancaster Stemmer', 'WordNetLemmatizer', 'SnowballStemmer' 'Word Tokenizer'. An advantage of using Lemmatization is, it considers the context of the sentence instead of a robust approach its counterpart Stemmer performs, thus it considers a few words in their root form as shown below.

WordNetLemmatizer: 'kind conservatives wanna everyone left communist antifa members', OFF
Lancaster Stemmer: 'kind conservatives wanna everyone left communist antifa memb', OFF

4.3 Data Filtering

To carry out Data Filtering, we previously removed stop words present in the NLTK stop word corpus. Now, we have added WordCloud corpus to tweak model precision. Also, words with length more than three are taken into account for both methods. Further, there is a change in value while considering the frequency of words across the dataset. Using previous minimum frequency value i.e. '20', we recovered less than 2000 unique word vector from possible 39450, thus resulting in the model to train on less amount of data. Hence we have changed the minimum frequency to '3' which generated more than 8000-word vectors. Figure 4, 5 and 6 depicts a WordCloud for subtask A, B, and C.

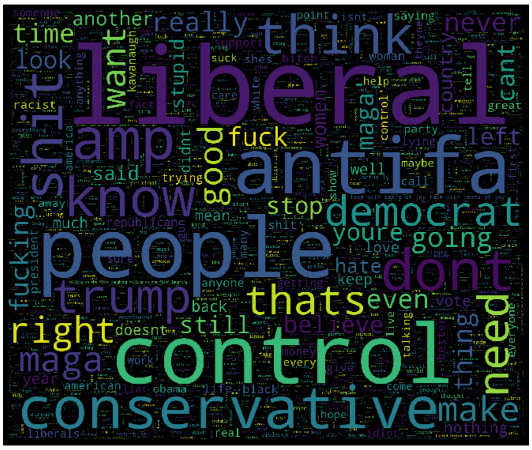

Figure 4: WordCloud for Subtask A

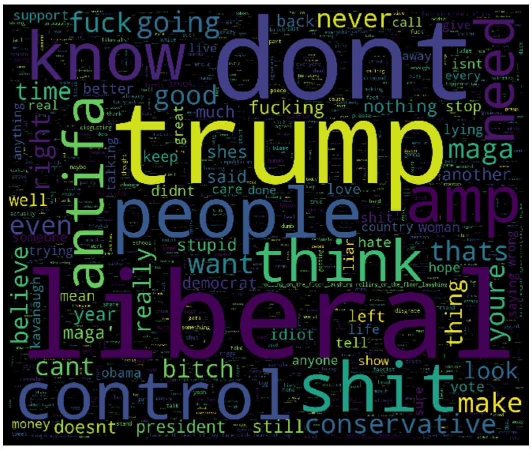

Figure 5: WordCloud for Subtask B

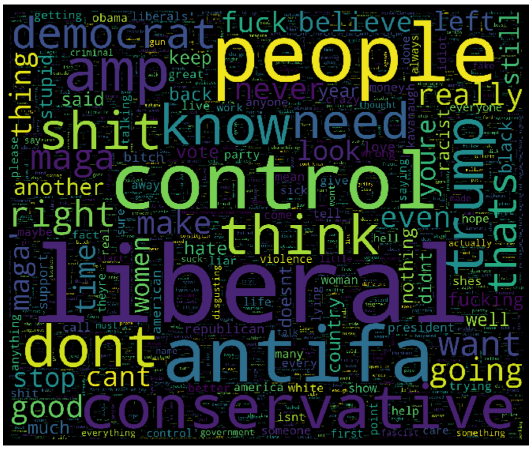

Figure 6: WordCloud for Subtask C

5. Modeling

For the following 3 models, we use StringToWordVector as the filter. Besides that, we use NGramTokenizer with N-gram max size=3 and N-gram min size=1. The classifiers were trained on the two different training set and tested on the corresponding two supplied test sets that are Lemmatized Dataset and Lancaster Dataset.

5.1 BayesNet

As for the first model, we use BayesNet with default parameters as the classifier. BayesNet is a probabilistic graphical model that represents a set of variables and their conditional dependencies through a directed acyclic graph (DAG). Bayesian networks are well-suited to deal with events that have already occurred and predict the likelihood that any of several possible known causes are contributing factors (Ben-Gal, 2007). The results are shown in Table 4.

Accuracy/Subtask	Dataset	BayesNet
Subtask_A	Lemmatized	79.5349
	Lancaster	79.6512
Subtask_B	Lemmatized	88.3333
	Lancaster	88.3333
Subtask_C	Lemmatized	58.216
	Lancaster	58.216

Table 4: Accuracy (%) for each subtask using BayesNet classifier

5.2 NaiveBayes

For the second model, we use NaiveBayes with default parameters as the classifier. NaiveBayes is a classification technique. In short, the naive Bayes classifier assumes that the existence of a particular feature in a class is irrelevant to the existence of any other features. NaiveBayes is good at building and is particularly useful for very large data sets. In addition to simplicity, Naive Bayes outperforms particularly complex classification methods (Kaviani, Pouria & Dhotre, Sunita, 2017). The results are shown in Table 5.

Accuracy/Subtask	DataSet	NaiveBayes
Subtask_A	Lemmatized	73.1395
	Lancaster	72.907
Subtask_B	Lemmatized	86.25
	Lancaster	86.25
Subtask_C	Lemmatized	56.8075
	Lancaster	57.7465

Table 5: Accuracy (%) for each subtask using NaiveBayes classifier

5.3 Sequential minimal Optimisation (SMO)

For the third model, we use Sequential minimal Optimisation(SMO(SVM)) with C1.0 PolyKernel. SMO is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support-vector machines (SVM) (Platt, John, 19998). SMO is a method of supervised learning. It is applied to statistical classification and regression analysis. The results are shown in Table 6.

Accuracy/Subtask	DataSet	SMO
Subtask_A	Lemmatized	80.814
	Lancaster	80.9302
Subtask_B	Lemmatized	88.75
	Lancaster	88.75
Subtask_C	Lemmatized	60.0939
	Lancaster	60.0939

Table 6: Accuracy (%) for each subtask using SMO classifier

6. Evaluation

For tasks A, B and C, we all use BayesNet, NaiveBayes, and SMO to classify the training set and use the prepared-test set to test the accuracy of our classifier. We also use True Positive(TP), precision and Recall to evaluate each model. The results are shown in Table 7.

Evaluation/Subtasks	Dataset	Model	Accuracy (%)	True Positive (TP)	Precision	Recall
Subtask_A	Lemmatized	BayesNet	79.5349	0.795	0.801	0.795
		Naïve Bayes	73.1395	0.731	0.696	0.731
		SMO	80.814	0.808	0.800	0.808
	Lancaster	BayesNet	79.6512	0.797	0.799	0.797
		Naïve Bayes	72.907	0.729	0.691	0.729
		SMO	80.9302	0.809	0.803	0.809
Subtask_B	Lemmatized	BayesNet	88.3333	0.883	0.787	0.883
		Naïve Bayes	86.25	0.863	0.835	0.863

		SMO	88.75	0.888	?	0.888
	Lancaster	BayesNet	88.3333	0.883	0.787	0.883
		Naïve Bayes	86.25	0.863	0.835	0.863
		SMO	88.75	0.888	?	0.888
Subtask_C	Lemmatized	BayesNet	58.216	0.582	?	0.582
		Naïve Bayes	56.8075	0.568	0.516	0.568
		SMO	60.0939	0.601	0.547	0.601
	Lancaster	BayesNet	58.216	0.582	?	0.582
		Naïve Bayes	57.7465	0.577	0.527	0.577
		SMO	60.0939	0.601	0.546	0.601

Table 7: Evaluation results for each model

High precision means that a classifier returned substantially more relevant/offensive tweets than irrelevant ones, while high recall means that a classifier returned most of the offensive tweets in the set.

From Table 6.1, we can learn the best classifier for subtask_A is SMO.
For subtask_B, the best classifier is SMO.
For subtask_C, the best classifier is SMO.

Using different training and test dataset would slightly influence the final result, but in most cases, the results do not change when using the same model but different datasets.

7. Conclusion

In this report we presented the results of the findings of the SemEval Offensive Language Identification in social media. In this task we used a dataset (Zampieri et al., 2019), which contains about 14000 tweets categorized into three layer hierarchical annotated model. The model consists of three Subtasks and the three subtasks are (A); whether the tweets are offensive (B); whether the offensive tweet is targeted or not (C); If the tweet is targeted so, is it targeted towards an individual, a group or others.

In total we used six different techiques of cleaning the data and prepared different models accordingly but in this report we only mentioned the best performing cleaning techniques and classifier midels to ealuate the offensive tweets and most of them are Supervised machine learning algorithms which are explained in the Modelling and Evaluation sections of the report.

8. References

[1] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In Proceedings of the 13th International Workshop on Semantic Evaluation,pages75–86, Minneapolis, Minnesota, USA. Association for computational linguistics.

[2] Ben-Gal I., Bayesian Networks, in Ruggeri F., Faltin F. & Kenett R., Encyclopedia of Statistics in Quality & Reliability, Wiley & Sons (2007).

[3] Platt, John (1998). "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines".

[4] Nikil Prakash, Dr. A. Aloysius. INTERNATIONAL EDUCATIONAL APPLIED RESEARCH JOURNAL (IEARJ) Volume 03, Issue 07, July 2019 E-ISSN: 2456-6713: DATA PREPROCESSING IN SENTIMENT ANALYSIS USING TWITTER DATA.

[5] Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.