OffensEval: Identifying and Categorizing Offensive Language in Social Media Research Workshop Report

Jiaqi Wang , Xin Wei , Xinyu Fan , Zixuan Tang
Group name: DDMC

Abstract

This paper presents our methods and models for subtask A, B and C of the Identifying and Categorizing Offensive Language in Social Media (OffensEval). The tasks are based on the Offensive Language Identification Dataset (OLID), which contains over 14,000 English tweets. The aim of subtask A is to identify offensive tweet. The aim of subtask B is to identify whether offensive tweets are targeted threat, targeted insult or untargeted. The aim of subtask C is to identify the type of objective that offense is targeted to. We investigated several models and get 82.2% accuracy of subtask A, 90.1% accuracy of subtask B and 66.7% accuracy of subtask C.

1 Introduction

Offensive language in social media such as Twitter, WeChat, is becoming a significant problem in recent years. Because of anonymous and big data volume of the Internet, offensive language could lead to serious bully problem. However, identifying offensive language manually is very time consuming and also cause pessimistic impact on the workers. So, developing effective autonomous methods to handle this problem is becoming widely concerned for researchers in Natural Language Processing and related field. For example, hate speech (Davidson et al. 2017), cyberbullying (Dinakar et al., 2011) and aggression (Kumar et al., 2018).

OffensEval addressed three subtasks related to offensive language based on Offensive language Identification datasets (OLID). There are three parts of OLID datasets for three subtasks respectively. Subtask A is about offensive language identification. Subtask B is based on subtask A which is automatic categorization of offense types. Subtask C is about offense target identification.

We have developed models for all of three subtasks of OffensEval. We used machine learning models including support vector machine (SVM), decision trees, linear regression and logistic regression. We also experimented on other methodology such as naïve Bayes, sequential minimal optimization (SMO). We used filters and emoji python package to clean the datasets.

A reminder of the paper: Section 2 discusses business understanding of OffensEval tasks. Section 3 discusses data quality and features of OLID datasets and tweets. Section 4 and 5 discusses our methodology for data preparation and modelling. Section 6 revises and evaluates our research and model performance.

2 Business Understanding

Business Understanding is the first stage of the

CRISP-DM process.

Project objectives and requirements

Offensive language is everywhere on social media. Individuals often attack others anonymously online via computers. Online communities, social media platforms and technology companies have been investing in research on how to deal with offensive language to prevent abuses on social media.

The task of this project is Identifying and Categorizing Offensive Language in Social Media. In this project we divide this total task into three sub-tasks.

- ✧ Sub-task A - Offensive language identification
- ✧ Sub-task B - Automatic categorization of offense types;
- ✧ Sub-task C - Offense target identification.

The business success criterion of the project is to identify whether it is offensive language and give the corresponding type of offense.

Data mining problem definition

The business goal of this project is identifying and categorizing offensive language in social media. So, data mining goal is organizing the data from the data source, then identify offensive language through write some Python codes and category offensive types the organized data through Weka, finally use different classification models to count the correct classification instance and correct classification rate.

Data mining success criterion of the project is to obtain correct offensive type and count the correct classification instance and correct classification rate.

Produce project plan

Next, this project will be divided into the following stages: data understanding, data preparation, modeling, evaluation, and improvement. Sort the data in the data source is important. The unsorted data cannot be directly imported into Weka, it need to sort the data through a piece of Python code. For the classified data, different models are needed to test the correct classification rate.

The data mining tool used in this project is Weka. It provides comprehensive support for the entire process of data mining, including preparation of input data, statistical evaluation of learning programs, visualization of input data and learning effects. ARFF is a Weka-specific file format, namely Attribute-Relation File Format. The Weka system includes all methods to deal with standard data mining problems: regression, classification, clustering, association rules, and attribute selection. The file is an ASCII text file that describes a list of instances that share a set of attribute structures. It consists of independent and unordered instances. It is a standard method for representing data sets in Weka. ARFF does not involve the relationship between instances.

3 Data Understanding

According to CRISP-DM, the second stage is data understanding which needs to acquire data list in the CSV and TSV files. But these files' format seems could not be read in Weka software. Thus, we need to clean these files and integrate these together. We need to check the attributes of the required data, and then predict the target attributes required by the task, the distribution of relationships, and perform simple statistical analysis. These analyses are helpful for data understanding and data preparation. Finally, we need to check the quality of the data, that is,

A	B	C
OFF	TIN	IND
OFF	TIN	OTH
OFF	TIN	GRP
OFF	UNT	—
NOT	—	—

Picture 1: Three labels in OLID

whether the data is complete? Is the data correct?

Tweet	A	B	C
@USER She should ask a few native Americans what their take on this is.	OFF	UNT	NULL
#MAGA @USER 🎶 Sing like no one is listening…	NOT	NULL	NULL
@USER Figures! What is wrong with these idiots? Thank God for @USER	OFF	TIN	GRP
@USER Fuk this fat cock sucker	OFF	TIN	IND

Table 1: Four tweets from the OLID dataset, with their labels for each level of the annotation model

The main dataset used to train this task is OLID Zampieri et al. (2019), which we would explain in this section. The five languages included in OLID are Arabic, Danish, English, Greek and Turkish. For English we will run sub-tasks A, B and C. The dataset contains English tweets annotated using a hierarchical three-level annotation model. It contains 14553 annotated tweets divided in a training partition containing 13240 tweets and three test partitions containing 860, 240 and 213 tweets. The three different labels in OLID is shown in Picture 1.

Finally, four examples of annotated instances in training dataset are presented in Table 1.
Weka can only recognize meaningful words and sentences, but there are a lot of meaningless punctuation marks and signs, urls and unrecognized strings in the database, which could be seen in picture 2. At the same time, we need to expand the abbreviated words and convert uppercase letters to lowercase letters, if not, this will make the sentence ambiguous and even make the sentence uncomfortable. In addition, in social media, some words are changed one or two characters to escape the offense detection systems. For example, ‘fuck’ may be written as ‘f**k’, etc. Thus, it is necessary to having a list of English offensive words, that we can use this list to classify these tweets initially. However, there are words that, although not insulting, are combating insults, which are still classified as insults.

4 Data Preparation

The data preparation consists of (1) converting file formats into Arff and (2) cleaning datasets. The format of given datasets is CSV or TSV but they are not easy for Weka to process. In our project we use google Colab file function to upload original datasets. Using pandas and csv python library to obtain contains of CSV and TSV files and storing them into new Arff files. Then adding relation and attribute statements at the head to make Arff files usable. We also divided the training dataset into three Arff files

Pre-processing Results
@USER She should ask a few native Americans what their take on this is.
@USER @USER Go home you’re drunk!!! @USER #MAGA #Trump2020 👊US👊 URL
@USER Someone should'veTaken" this piece of shit to a volcano. 😂"
' she should ask a few native americans what their take on this is'
' go home you are drunk maga trump oncoming fist united states on coming fist '
' someone should have taken this piece of shit to a volcano face with tears of joy'

Table 2: Tweets before and after pre-processing

with labels for subtask A, B and C respectively to cope with the three test datasets. We cleaned all datasets before using StringtoWordVector to get word vectors. In social media, there are some methods that could convert words in a way to be ignored by the offense detection system. For instance, 'asshole' could be written as 'a**hole', 'a$$hole', etc. We searched English offensive words online and created several filters to convert symbols into words. Some tweets in OLID datasets use @USER symbol to mention person or entity, so we replace multiple @ symbols with single @. OLID datasets also use URL symbols to replace the link to a website or other urls in tweets. So, we replace all URLs because we could not get information from these symbols. We also removed all punctuation and symbols from the tweets since it is difficult to analyse offensive emotion from them. All abbreviations are expanded and all words are converted to lowercase as it could reduce the searching space and help to understand the meaning of tweets. We use python emoji package to decode every emoji into a short sentence to describe them, so that we could keep their sentiment and use them to analyse the sentiment of respective tweet. The converted emoji words will also be processed by the filters we described before.

After we assured that the datasets are clean, we applied the datasets to StringtoWordVector filter to convert tweet string to word vector. We set the minimum term frequency to 5 to remove the infrequent words. We also applied different stemmer methods to reduce searching space and tokenizer to encode the word list.

5 Modelling

In the section of data modeling, we deal with tasks A, B and C respectively.

In the initial solutions:

Training set:

Sub-task A: Offensive language identification:

Model	Accuracy
Bayes	**0.7421**
SGD (SVM)	0.7419
SGD (logistic regression)	0.7327

Sub-task B: Automatic categorization of offense types:

Model	Accuracy
Bayes	0.8132
SGD (SVM)	0.7973
SGD (logistic regression)	**0.8352**

Sub-task C: Offense target identification:

Model	Accuracy
Bayes	0.6375
ZeroR	0.6210
WeightedInstancesHandlerWrapper	0.6210
FilteredClassifier	**0.6460**

Testing set: (Tasks performance of different models on the test set.)

Sub-task A: Offensive language identification:

Model	Accuracy
Bayes	0.7884
SGD (SVM)	0.7453
SGD (logistic regression)	**0.7558**

Sub-task B: Automatic categorization of offense types:

Model	Accuracy
Bayes	0.8333
SGD (SVM)	0.8458
SGD (logistic regression)	**0.8542**

Sub-task C: Offense target identification:

Model	Accuracy
Bayes	0.5728
ZeroR	0.4695
WeightedInstancesHandlerWrapper	0.4695
FilteredClassifier	**0.6385**

In the final solutions:

Sub-task A: Offensive language identification:

Model	Factor	Accuracy
Bayes	default	0.7443
	useKernelEstimator: True useSupervisedDiscretization: True	0.8126
SMO	**PolyKernel**	**0.8222**
	Puk	0.7745

Sub-task B: Automatic categorization of offense types:

Model	Factor	Accuracy
Bayes	**default**	**0.9012**
	useKernelEstimator: True useSupervisedDiscretization: True	0.8752
SMO	PolyKernel	0.8957
	Puk	0.8876
J48	default	0.8707

Sub-task C: Offense target identification:

Model	Factor	Accuracy
Bayes	default	0.6291
	useKernelEstimator: True useSupervisedDiscretization: True	0.6478
SMO	PolyKernel	0.5776
	Puk	0.4649
J48	**default**	**0.6667**

For subtask A, in the original solution, we used Bayes, SGD (SVM) and SGD (logistic regression), and got the accuracy of 0.7421, 0.7419 and 0.7327, respectively. Among them, Bayes was the best. In the final solution, firstly, the Bayes classifier of the two parameters were modified by me, one is set useKernelEstimator to True, another is set useSupervisedDiscretization to True, then the accuracy of 0.8126, the accuracy increased relative to the default parameters. Secondly, we used the SMO classifier and set the kernel parameters to PolyKernel and Puk respectively, and the accuracy was 0.8222 and 0.7745 respectively. SMO with the PolyKernel is the best.

For subtask B, in the original solution, I used Bayes, SGD(SVM) and SGD (logistic regression), and the accuracy was 0.8132, 0.7973 and 0.8352, respectively. Among them, SGD (logistic regression) is the best. In the final solution, firstly, we modified the Bayes classifier with two parameters: one was set to True for the useKernelEstimator and the other was set to True for the useSupervisedDiscretization, with an accuracy of 0.8752, which was improved relative to the original solution. Secondly, the accuracy of the default value of Bayes is 0.9012. Then, we used the SMO classifier to set the kernel parameters to PolyKernel and Puk, and the accuracy was 0.8957 and 0.8876, respectively. Finally, the default value of J48 is 0.8707. Bayes with the default factor is the best.

For subtasks C, in the original solution, we use the Bayes theorem, ZeroR, Weighted InstancesHandlerWrapper and Filtered Classifier, the accuracies were 0.6375, 0.6210, 0.6210 and 0.6460. Among them, FilteredClassifier (classifier: J48 and filter: StringToWordVector) is the best. In the final solution, we first modified the Bayes classifier and set two parameters: one parameter is set to True for useKernelEstimator, and the other parameter is set to True for useSupervisedDiscretization, with an accuracy of 0.6478, which is not much different from the original scheme. Second, the accuracy of the default value of Bayes is 0.6291. Then, we used SMO classifier to set the kernel parameters to

PolyKernel and Puk, and the accuracy was 0.5776 and 0.4649 respectively. Finally, the default value for J48 is 0.6667. J48 with a default factor is best.

6 Conclusion

Offensive language identification is a significant task for online social media. We did researches on previse works and methods before introducing our models to the three OffensEval subtasks including SVM, linear regression and SMO. We cleaned OLID datasets and used Weka to build and test our models and found various results. The judgment used for evaluation are accuracy.

References

Bogdan Lazarescu, Christo Lolov, and Silvia Sapora. 2019. *OffensEval at SemEval-2019 Task 6: Okham's Razor on Identifying and Categorizing Offensive Language in Social Media.* https://arxiv.org/abs/1903.05929v3

Doostmohammadi, Ehsan and Sameti, Hossein and Saffar, Ali. 2019. Ghmerti at SemEval-2019 Task 6: A Deep Word- and Character-based Approach to Offensive Language Identification. In *Proceedings of the 13th International Workshop on Semantic Evaluation,* page 617-621. https://www.aclweb.org/anthology/S19-2110.

Vyshnav M T, Sachin Kumar S, and Soman K P. *Offensive Language Detection: A Comparative Analysis.* https://arxiv.org/abs/2001.03131

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, and Noura Farra, Ritesh Kumar. 2019. *SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval).* https://arxiv.org/abs/1903.08983

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. *Automated hate speech detection and the problem of offensive language. In Proceedings of the International Conference on Weblogs and Social Media (ICWSM).*

Karthik Dinakar, Roi Reichart, and Henry Lieberman. 2011. *Modeling the detection of textual cyberbullying. In The Social Mobile Web, pages 11–17.*

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. *Benchmarking aggression identification in social media. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC).*

Ehsan Doostmohammadi,,Hossein Sameti, and Ali Saffar. 2019. *Ghmertiat SemEval-2019 Task 6: A Deep Word- and Character-based Approach to Offensive Language Identification.*

Amir HOSSEIN Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. *Offensive Language Detection Using Multi-level Classification*.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 shared task on *the identification of the GermEval 2018 shared task on the identification of offensive language.*

Shearer C., *The CRISP-DM model: the new blueprint for data mining*, J Data Warehousing (2000); 5:13—22.

Meta S. Brown. 2014. *Data Mining For Dummies. ISBN: 978-1-118-89316-6*
https://www.dummies.com/programming/big-data/phase-1-of-the-crisp-dm-process-model-business-understanding/

Witten, Ian H.; Frank, Eibe; Hall, Mark A.; Pal, Christopher J. 2011. *"Data Mining: Practical machine learning tools and techniques, 3rd Edition"*. Morgan Kaufmann, San Francisco (CA). Retrieved 2011-01-19.

Holmes, Geoffrey; Donkin, Andrew; Witten, Ian H. 1994. "*Weka: A machine learning workbench*" (PDF). Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia. Retrieved 2007-06-25.