Bil476 fall 2020 Project

Monitoring Suspicious Discussions On Online Forums

Oğuz Kaan Özsoy
o.ozsoy@etu.edu.com
171101061

1 Abstract

Now a days using the internet as a means of conversation is a must for most. As internet technology has been growing more and more, both legal and illegal activities have contributed to this technology. It appears that a lot of first hand news are covered in Internet forums long before conventional mass media has a grasp on it. Internet has a lot of channels for communication which provides an efficient feedback for illicit activities such as copyrighted film distribution, threatening messages, online gambling, sexism and toxic behaviour. It is impossible for humans alone to make sense of this data thus using techniques of data mining will be most of use.

2 Brief Description of The Project

In this project, I will be collecting number of data from open forums. Using a variety of techniques to extract information out of the websites such as web scraping, using up to date databases and online data sources. Then I will be using data mining techniques on this set of collected data to classify suspicious and toxic text from non-suspicious text. As a result we will have access to more refined information like percentage of suspicious conversations on a target website. And a caparison between websites using this classification and scraping techniques.

3 Dataset Description

I have collected comment data from various reddit pages. This data included page name, topic, username and comment. I hand forged few of the data myself for early tests. Finaly I used a toxic tweet data from github by t-davidson(https://github.com/t-davidson/hate-speech-and-offensive-language). I formated the data into partitions for my classification algorithms to use. For the imbalanced data I managed a balanced csv file for future classification tasks.

4 Crawling

I used scrapy for crawling with a constructed reddit spider. Since reddit was a hugely famous forum, it had all kinds of controversial forum pages which included suspicious and toxic comments and discussions. Fallowing pages bellow were my entry points to the web site. Spider had simple instructions to fallow next page links and get the comments on current page.

'https://www.reddit.com/r/leagueoflegends/comments/'
'https://www.reddit.com/r/lgbt/comments/'
'https://www.reddit.com/r/Conservative/comments/'
'https://www.reddit.com/r/ToxicAMWF/comments/'
'https://www.reddit.com/r/atheism/comments/'
'https://www.reddit.com/r/Feminism/comments/'
'https://www.reddit.com/r/unpopularopinion/comments/'

Figure-1: Entry Points To Forum Discussions

5 Spark NPL Classification

Spark NLP is a Apache Spark supported Natural language processing library that's base language is in scala. In this project I used it's python API. Promise is you can built a pipeline for Spark NLP to use to train your model and it will take care of it using it's deep learning solutions. Fist I tried simple classification techniques on my data using Spark NLP. To test the framework I wrote a classifier that classifies if a text belongs to a forum. On test benchmarks It got about 90% accuracy. This just shows how powerfull this library is. It has number of annotators that can be placed in the pipeline to focus on a subject. One of the annotators is a pretrained model that checks for spelling mistakes. The main business happens in an annotator called word embeddings which is also called universal word embeddings(use) . This step converts the sentences into learning usable vectors.

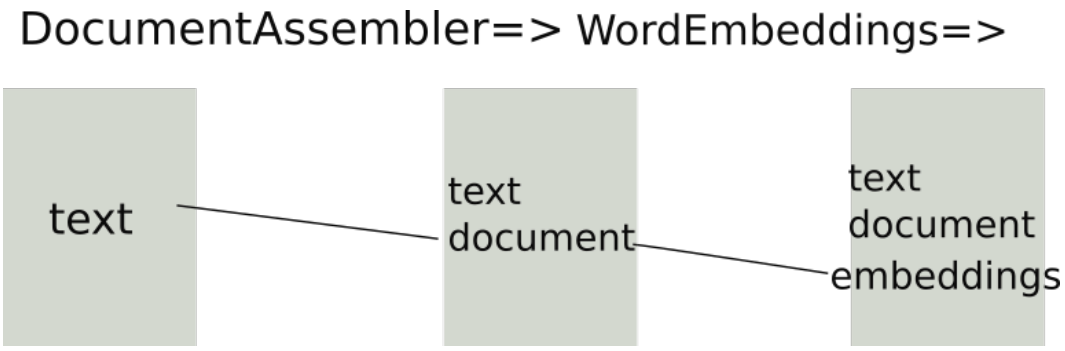

Figure-2: Example Pipeline

After acquiring the vectors, these vectors are fed to Sparks own Deep learning model for training. I did add a sentence detector, tokenizer, normalizer, cleaner and a lemmatizer to its pipeline. Except those only parameter I did change was setting number of epochs to 20 since it worked quite fast. But sinced classes were unbalanced spark didn't work quite as successful until I forged a csv file with class count equal for all classes(There were 3 classes in the database I depended training on).After training on test scores Spark NLP got an accuracy of 84% on suspicious text classification.

6 Spacy and Tensorflow Neural Network Classification

Since Spark NLP had it's own Deep learning model implemented inside, I wanted to build a neural network of my own to compare the results using word embeddings vectors of Spacy library. I used a word embedding called "en_core_web_lg" which helped me produce 300 sized vectors for each sentence. I used tensorflow for my machine learning tasks to build a fully connected feed forward network. My architecture is given bellow figure.

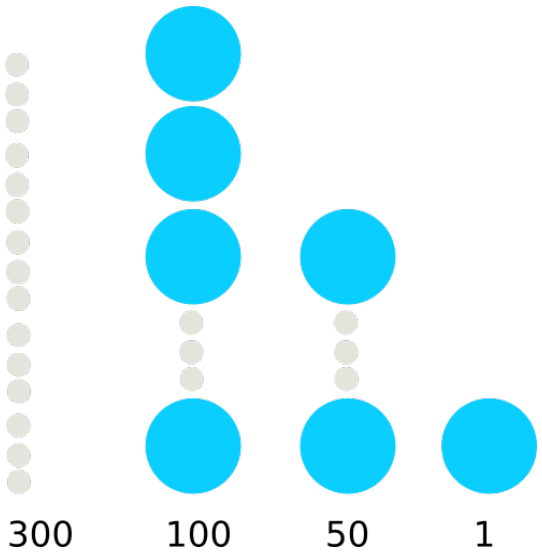

Figure-3: Neural Network Architecture

As can be seen from **Figure-3** this network consist of four layers. First layer is just an input layer which takes all float values in the vector constructed by Spacy embeddings. First two dense layers use the relu activation function while last one uses sigmoid as output to get values between 0 and 1 as output. To combat over fitting I used lasso and ridge regularization on each dense layer with minimal weight of 0.001. As loss function I used Binary Cross entropy function as it is good for binary classification. I run this model for 10 epochs on all the data I have. After training on test scores, my neural network got an accuracy of 90% on suspicious text classification. But later when I hand checked the results with Spark NLPs results it was nearly as powerfull if not 90%.

7 Conclusion

Finally I saved all the classified text data into csv files then imported them to my Postgresql database. It was easy to compare results of tensorflow and Spark NPL classification side by side on Postgre admin. Both tensorflow and Spark NPL classifiers classified similar sentences. This was another method measuring success rate of my tensorflow model. As result we were able to classify text with high success rate. Some of the suspicious text needed to be hand inserted but thanks to t-davidson's offensive tweet data. I was able make two classifiers.

8 References

- Github:https://github.com/AlfaPigeon/Suspicious-and-Toxic-Discussion-Classification-on-Online-Forums
- Libraries: https://scrapy.org/ https://www.tensorflow.org/ https://nlp.johnsnowlabs.com/ https://spacy.io/ https://numpy.org/
- Article on Spark:https://towardsdatascience.com/introduction-to-spark-nlp-foundations-and-basic-components-part-i-c83b7629ed59
- t-davidson's offensive tweet data: https://github.com/t-davidson/hate-speech-and-offensive-language
- Reddit pages: https://www.reddit.com/ 'https://www.reddit.com/r/leagueoflegends/comments/', 'https://www.reddit.com/r/lgbt/comments/', 'https://www.reddit.com/r/Conservative/comments/', 'https://www.reddit.com/r/ToxicAMWF/comments/', 'https://www.reddit.com/r/atheism/comments/', 'https://www.reddit.com/r/unpopularopinion/comments/', 'https://www.reddit.com/r/Feminism/comments/'