International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May-2021 ISSN: 2582-3930

Automation in Social Media Comments with Robust Fast-Text and CNN

Shelake S.D.[1], Rokade M.D.[2]

[1]*PG Student, Department of Computer Engineering, SPCOE Otur, Pune, Maharashtra, India*
[2]*Assistant Prof., Department of Computer Engineering, SPCOE Otur, Pune, Maharashtra, India*

Abstract -*Social media networking and conversation in that, online conversation platforms give us the power to share our opinions and ideas. However, nowadays on social media platforms, many people are taking these platforms for pride, they see cyber-attacks and cyber-bullying [3] as an opportunity to harass and target others which in extreme cases leads to traumatic experiences and suicide attempts. Is. Manually identifying and categorizing such comments is a very long, tedious and unreliable process. To address this challenge, we have developed a deep learning methodology [1] that will identify such negative content on the discussion online discussion platform and successfully categorize them into appropriate labels. The main goal of our proposed model is to implement the invented text-based convolution neural network (CNN) with word embedding, using Fast-Text word embedding technology. Fast-Text has shown efficient and more accurate results compared to the Word2Vec and Glove models. Our model mainly focuses on to improve the detection of different types of toxins to improve the social media experience. Our model Dell classifies such comments into six categories: toxic, serious toxic, obscene, threatening, insulting, and hateful. Multi-label classification helps us provide automated solutions to the problem of incoming toxic comments.*

Key Words:Social media, Fast-Text, CNN, GLOVE, TFIDF.

1.INTRODUCTION

In today's era, social media is one of the most common and easy ways to communicate and express one's thoughts. This platform allows discussion on various topics, sharing information and opinions on the topic. But nowadays it can be difficult to maintain etiquette and good manners or behavior on these platforms. Offensive content, harassment, a lot of jeering, cyber-resistance related activities such as platforms that have a detrimental effect on a person's mental and psychological health have become very common. This can sometimes lead to harmful and lifelong traumatic effects on a person. This type of situation can shock users and lead them to express their opinions, completely eliminate themselves and stop seeking and receiving help from others. Companies owned by such discussion online discussion platforms are working on various solutions such as comment classification technology, user blocking method and comment filtering systems. In the comment classification approach, [4] the goal is to categorize comments or sentences into different categories based on their toxicity level [5]. By categorizing these comments, the team can take appropriate steps to prevent the occurrence and growth of negative influences created by such activities on social platforms. Such a multi-label taxonomy model would make the purpose of social communication on social media more effective and positive [3]. By automating this comment classification approach, companies can save their time and also the manual efforts to moderate these platforms. The data we used for our model is Kaggle's toxic comment classification dataset on Wikipedia's talk page edits. CNN (Convolution Neural Network) using, our goal is to develop a multi-label classification model that categorizes comments into 6 different categories: toxic, serious toxic, obscene, threatening, insulting, and hate based on its toxic level [7].

2. OBJECTIVE

In this, section going to define the main purpose and goal of system which needs to keep in mind while implementation.

- Word Create a word embedding mechanism that can help identify vulgar or negative terms in a comment.
- Class Identify the class of slang based on the level of toxicity and assign the corresponding weight to it.
- Perform an efficient pre-processing unit to make the data suitable for data analysis
- Create a CNN model to process the classification by training the model with training data fit and test the model to evaluate the accuracy rate.
- Modify the model through back-propagation mechanism and manage the trade-man in between Over-fitting and under-fitting.

Therefore these are the objective which initially helps to understand the problem statement working model of system.

3. PROPOSED SYSTEM

Our proposed model is a system made up of Fast-Text word embedding technic, by this technology and CNN (Convolution Neural Network) [2] that do multi-label classification of toxic comments [6]. In this system, input as a comment will be given from social sites which will be analyzed and sent to the word embedding stage. At this stage, sentences are broken down into words and embedded in vectors that are processed by CNN. After evaluating the comment by a trained CNN model, the resulting labels or categories of toxicity will be inferred and visualized.

The architectural design of our proposed model, which represents the software architecture, is shown in Figure 1 and the work-flow model is shown in Figure 3 below.

In the architecture, representing the different modules technique with its works that is how online platform, word

embedding and CNN model is working collectively to achieve the system goal.

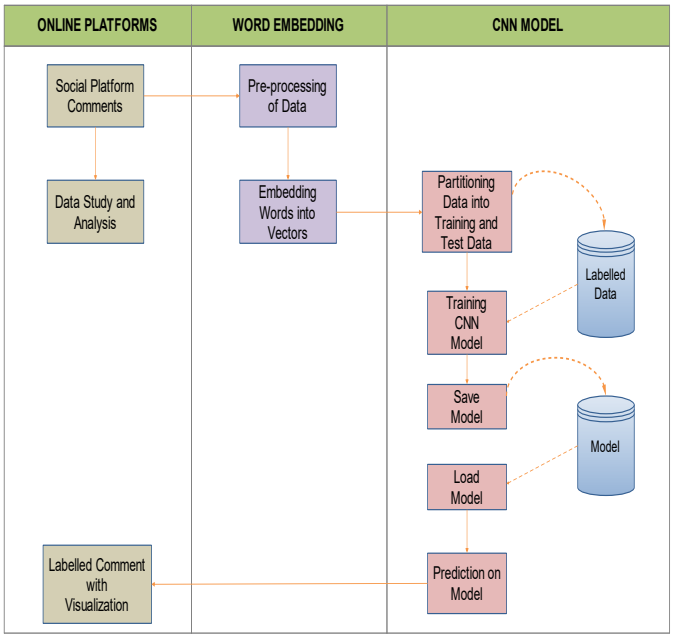

Fig -1: System Architecture

1. Comments from social media platforms: Our gathering of comment data from desired sources.

2. Study of Data and Analysis: The task is to analyze the nature of the data collected, and try to visualize the data present in terms of the number of words and labels provided in the training group. Figures 1 show the visualization done in our project with the Wikipedia dataset. We can be found that the number of comments in our dataset is less than 200 words. Also, we can say an issue from that most of the comments belong to the categories of toxic, [6] [8] obscene and insulting.

3. Data Pre-Processing: To prepare the data for the model, we remove all punctuation, lowercase letters, stop-words, and the addition of numeric digits.

4. Embed word in vector: Processed comments are converted to vector format in numeric representation using Fast-Text.

5. Divide into training and test data set: Divide data into training and test data with a ratio of 70:30, respectively.

6. CNN Training: First layer convolution on embedded word vectors using multiple filter sizes, then pooling layers and gala layers, the model will be saved [2].

7. Save Model: The saving model is of contain checkpoints for future predictions. This also helps when tuning or adjusting the parameters of the model to improve accuracy

8. Load model: Load model for predictive guess on new unseen data, i.e., or unseen or invisible comments.

9. Visualizing with Labels: Visualizing or displaying tagged or labeled comments at the end of the platform Efficient visualization of the results with telling a good story of data or insights can help organizations take appropriate action for their platform and minimize the likelihood of such activities occurring. The effect is also created.

4. METHODOLOGY

Deep learning algorithms [1] or machine can't process strings that is plain text as an inputs [3]. These algorithms are unable to process strings such as raw input. The word embedding technique provides a solution to this issue by converting string text into a numeric format or vector format that can be used by the model and can also be used to find the semantic relationship between the related words by calculating the differences between the two. Vector, known as embedding space. Our proposed model is using Fast-Text as a word embedding technique. Fast-Text uses n-gram characters as a lowercase unit. For example, the vector word 'apple' can be broken down into different word vector units ["ap", "app", "le", "ple"]. So, the embedded vector word for "apple" is the sum of all these n-grams. The advantage of Fast-Text is that it produces better embedding words for rare words, or not even words seen during training because the vectors of the N gram character are shared with other words.

This is one thing Word2Vec and the GLOVE (Global Vector for Word Representation), bacon can't bring home. This means that even for archaic words (e.g. due to typing errors), the model can make an educated guess about its meaning.

$$-\frac{1}{2}\sum_{n=0}^{\infty}(y_n \log(\mathrm{f}(\mathrm{BA}x_n)))$$

The Convolution Neural Network (CNN) is a feed-forward neural network consisting [2] of three layers, the compromise layer, the pooling layer, and the fully connected dense layer. CNN is mostly used for image classification work. Here, instead of an image, CNN's input is documents presented as sentences or metrics. In this, each row of the matrix can be a word or even a character where each row is a vector quantity that represents the word. These vectors are low dimensional representations of words or what we call embedding words like Word 2vec or GLOVE [2], but they can also be single-hot vectors that indicate the word in the vocabulary. For a 10-word sentence employing 100-dimensional embedding, our input could be 10×100 metrics. By the means of visually, our filters slide on local areas of an image, but in NLP we typically use filters that slide over entire rows of matrix (words). Thus, the "width" of our filters is usually equivalent to the width of the input matrix, height or region size may vary, although windows longer than 2-5 words at a time are typical. For the multi-label taxonomy function, CNN In, we will use the soft-max layer as the output layer, as it assigns the probability of a particular label in the range of 0 to 1.

$$\sigma(z)_j = \frac{e^z j}{\sum_{k=1}^{n}(e^z k)}$$

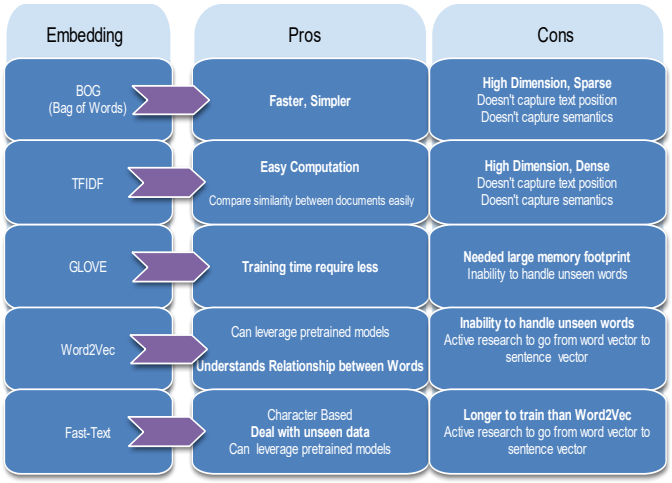

Fig -2: Word Embedding Technique: Comparative Study

The first layer of CNN embeds words into a low-dimensional vector. The next layer embedded word vectors involves multiplication on multiple filter sizes of abuse. For example, three at a time, or words slide over. Next, we will do max-pooling. The results are placed in a long feature vector that is given for dropout regularization, and the result is categorized using the soft-maximum level. For each word we have a look-up table for word embedding.

These M-dimensional embedding's start from square measure random and when the coaching is updated. The representations of this term averaged in the representations of the text after the square measure. After embedding, the embedded token is fed into the convolution and pooling layer and with a slight dropout, the objective can be achieved. Figure 2 represents the workflow of the classification task. In that done the comparison of different embedding technic with their pros and cons. The including techniques are BOW (Bag of Words),TFIDF (Text Frequency Inverse Document Frequency), GLOVE (Global Vector for Word Representation), Fast-Text and CNN (Convolution Neural Network) for the better understanding.

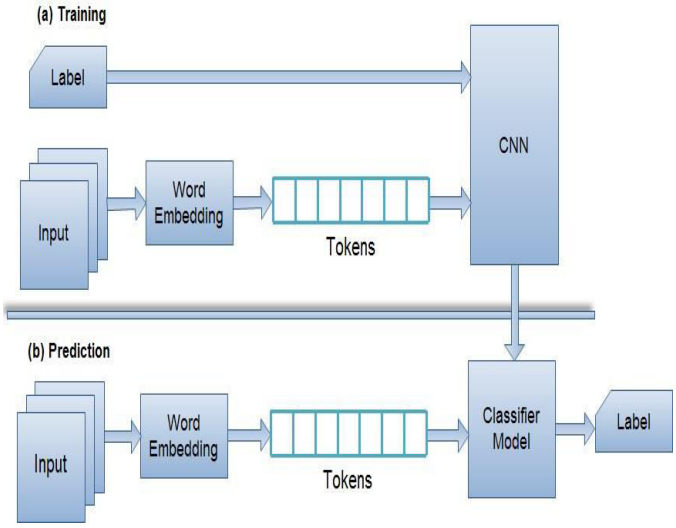

Fig -3: Model Implementation Workflow.

5. CONCLUSIONS

Our model, based on multi-label classification using Fast-Text and CNN that is using Convolution of the Neural Network method and is useful for detecting and classifying the toxic and offensive comments on social media platforms according to their toxicity. We've introduced multiple approaches to classifying comments which are toxic using Fast-Text and Word2Vec accordingly. Using the classification obtained here, social media platforms can implement this system and prevent negative influences on social media. As we tested between Fast-Text and Word2Vec and our results concluded that Fast-Text is more accurate and when slangs, jargons, typing errors and short forms are used. The model is especially out of shape when the data and the size of the dataset are large. Finally, we present the practical system applied with Fast-Text above the Word2Vec model with a definite result, which is interesting for researching toxic comments taxonomy of domain classification.

REFERENCES

1. MukulAnand and Dr.R.Eswari, "ClassificationofAbusiveComments in SocialMedia using DeepLearning," 3rd International Conference on Computing Methodologies and Communication (ICCMC), IEEEXplore, 2019 Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1109/ICCMC.2019.8819734.
2. Maryem Rhanoui, MouniaMikram, SihamYousfiandSoukainaBarzali, "A CNN-BiLSTM Model for Document-Level Sentiment Ànalysis ," Machine Learning and Knowledge Extraction, 2019, vol. 1, pp.832–847; doi:10.3390/make1030048.
3. Thedora Chu, Max Wang, Kylie Jue. "Comment Abuse Classification with Deep Learning", Stanford University, 2017.
4. KarthikDiankar, RoiRiechart and Henry Lieberman, "Modeling the DetectionofTextualCyberbullying." Massachusetts Institute ofTechnology, Cambridge MA 02139 USA.
5. KarthikDiankar, RoiRiechart and Henry Lieberman, "Modeling the DetectionofTextualCyberbullying." Massachusetts Institute ofTechnology, Cambridge MA 02139 USA.
6. Suresh Mestry, Roshan Chauhan, Hargun Singh, KaushikTiwari and Vishal Bisht , "Automation in Social Networking Comments With the Help of Robust fastText and CNN", 1st International Conference on Innovations and Informations in Communication Technology (ICIICT), IEEE Xplore, 2019
7. Xin Wang, Yuanchao Li, Chengjie Sun, Baoxum Wang and Xialong Wang., "Polarities of Tweets by Composing WordEmbeddings with Long Short Term Memory", 7th International Joint Conference of Natural Language Processing, July-2005.
8. ManavKohli, Emily Kuehler and John Palowitch, "Paying Attention to Toxic Comments Online", StanfordUniversity.