National College of Ireland

Using hybrid deep learning and word embedding based approach for advance cyberbullying detection

MSc Research Project
MSc. In Data Analytics

Jigar Bhatt
Student ID: x18179959

School of Computing
National College of Ireland

Supervisor: Prof. Paul Stynes

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Jigar Sanjay Bhatt

Student ID: X18179959

Programme: MSc. In Data Analytics **Year:** 2019-20

Module: Research in Computing

Supervisor: Prof. Paul Stynes

Submission Due Date: 17/08/2020

Project Title: Using hybrid deep learning and word embedding based approach for advance cyberbullying detection

Word Count: 7351 **Page Count:** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Jigar Sanjay Bhatt

Date: 17 - 08 -2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Using hybrid deep learning and word embedding based approach for advance cyberbullying detection

Jigar Sanjay Bhatt
X18179959

Abstract

The ever-increasing use of social media in the internet space have induced a number of problems like cyberbullying and cyberaggression over the internet. Researchers have made a commendable progress on the ongoing fight against cyberbullying but a lot of unresolved issues still persist that primarily motivates the purpose of the research. The paper aims to integrate recent advances in the field of word embedding like fastText, ELMo and stacked flair embeddings combined with a host of robust deep learning techniques to further the efficiency of detection over the state-of-art. Two distinct datasets Formspring and Wikipedia were requested and processed for the purpose of the research. A number of different combinations of word embedding with deep learning methods were tested and compared with CNN with ELMo embedding delivering the most promising results with an F1 score of 0.82 on both datasets. On the other hand, CNN with fastText obtained F1 score of 0.82 on Formspring and 0.64 on Wikipedia dataset but was computationally faster than the counterparts. Moreover, transfer learning was performed using the models to test and prove the robustness and efficacy of the models. The system performed considerably well with superior scores in precision, recall and F1 over the state-of-the-art across all the test cases performed.

1 Introduction

Today, with the ever-increasing use of social media, bullying is no longer just limited to school campuses or neighborhood but the entire internet space where the embarrassment of the insults or personal attacks can be noticed by millions worldwide. As per the statistics presented by the i-SAFE foundation, about 25 percent of teenagers and adolescents have been a prey to cyberbullying in one form or the other.[1] Many of the incidents also end up with victim's suicide. The alarming figures in itself indicate the magnitude of the problem and the potential effect it can have on the victims falling prey to cyberbullying. A considerable amount of research work has been dedicated in dealing with this sensitive issue but the mission of achieving an outright bullying free internet space is still not achieved hinting a significant scope for improvement. The reason that the issue hasn't completely resolved yet is due to highly intricate nature of human language interpretation. A number of detection

[1] http://www.bullyingstatistics.org/content/cyber-bullying-statistics.html

approaches have been experimented in the past that solely rely on a number of handcrafted features such as POS tagging, frequency of swear words used in a sentence. Such approaches fail to address the hidden nuisances present in human language and are often unsuccessful in accurately predicting the cyberbullying instances. (Agrawal and Awekar, 2018)

The task of detection of cyberbullying has considerably evolved with techniques ranging from simple classification models to complex hybrid models explored in the past. Lately deep learning has unfolded as a superior alternative to the traditional machine learning algorithms employed in the past. Deep Neural Network based models popularly used for image processing and computer vision have demonstrated a lot of promises to outperform the traditional machine learning based models in a lot of Natural Language Processing (NLP) tasks. As far as detection method is concerned, widely used state-of-the-art methods like Bag of Words (BOW), TF-IDF and so on focus primarily on the profanity of the language and fail to address the variation in the language. The recent advances in the field of Natural Language Processing (NLP) gave rise to dynamic language processing and detection method word embeddings. Word embeddings are a class of methods that assign numerical representation to the words in the vector space on the basis of cosine similarity between the words. (Zhao, Zhou and Mao, 2016) With the birth of word embeddings, a number of techniques like word2vec, GloVe, Bert and so on got introduced in the field of NLP. The striking feature of word embedding is that semantic representation of words in vector space enables a machine to actually understand the actual meaning of the sentence rather than treating as random string of numbers. (Al-Ajlan and Ykhlef, 2018)

Agrawal et al. in (Agrawal and Awekar, 2018) presented the use of GloVe and SSWE along with Deep Neural Network (DNN) based models in the detection of cyberbullying in the form of racism, sexism, personal attack and bullying. This shows the integration of word embeddings in the task of cyberbullying detection is quite promising. GloVe embedding developed by Stanford NLP group is one of the widely used word embedding techniques in NLP tasks. But this method focuses more on the syntactic information contained in the sentence and often fail to understand the underlying sentiment behind the message. Another major drawback of the technique is its inability to deal with the words outside the corpus. This paper aims to further the detection efficiency by incorporating a number of newly introduced detection techniques like fastText, ELMo and flair stacked embedding along with the robust DNN models. fastText embedding have the ability to generate embedding for words that are not a part of the training corpus. (Bojanowski *et al.*, 2017) ELMo and flair embeddings are context aware techniques that are quite impressive in understanding the sentimental information conveyed in the message rather than relying on individual words.(Akbik, Blythe and Vollgraf, 2018) For the purpose of training the data, DNN based models are used that are proficient in the task of classification minimizing the classification detection process workflow consisting of extra feature selection and feature extraction steps.

The study revolves around carrying out extensive research in the field of cyberbullying detection with the following research question as a prime motivation.

To what extent can DNN models integrated with advanced word embedding methods enhance the efficiency of detection of cyberbullying instances?

To address the research question, the following set of research objectives were derived:

1. Investigate the state-of-the-art in the field of cyberbullying and discuss the challenges and limitation associated with detection of bullying instances.
2. Design and implement comprehensive word embedding based DNN models with superior detection efficiency to overcome the limitations in the state-of-the-art.
3. Perform the cross-platform testing of the models to inspect the robustness of the models.
4. Thoroughly evaluate the performance of the models on the basis of evaluation metrics.

The rest of the paper is organized as follows: Section 2 will broadly discuss the previous literature and state-of-the-art in cyberbullying detection. Section 3 will present the research methodology followed in the project. Section 4 will discuss the system architecture and design. Section 5 will discuss the implementation of the model in details. Section 6 will discuss the evaluation and comparison of the proposed models. Section 7 will broadly discuss the conclusion, limitations and future work of the project.

2 Related Work

A lot of research work has been done in the field of cyberbullying detection. A number of different approaches ranging from simple statistical model to complex hybrid models have been experimented in the past in an attempt to detect cyberbullying and cyberattacks happening in the internet space. The following section aims to highlight and investigate the key previous work done in the filed of cyberbullying detection.

2.1 Statistical and Probabilistic detection

This subsection will look at some of the basic probabilistic and statistical methods employed in order to detect cyberbullying instances.

In one of the studies, (Zois *et al.*, 2018) came up with probabilistic cyberbullying detection framework. The framework focused on using probability computation and two-way hypothesis to identify potential traces of cyberbullying. In such similar study, authors proposed a fusion framework of textual, social and image-based features. Authors used probabilistic method to assign confidence scores on the basis of individual modalities (Singh, Huang and Atrey, 2016).

In another study, authors proposed the idea of using K-means clustering in order to classify the posts into one of the 8 categorical classes predefined by the authors. The model used for the purpose of clustering was Naïve Bayes classifier which used probabilistic determination to assign the individual post to one of the categories. (Romsaiyud *et al.*, 2017)

While the methods looked decent on the outer level, the models lack the intrinsic ability to detect the hidden nuances in the texts and would often fail when applied in real world data that is inherently tricky to interpret in nature.

2.2 Rule based and feature based detection with machine learning

The methods highlighted in this section mainly focused on feature based cyberbullying detection blocks combined with machine learning models.

In one of the studies by (Van Hee *et al.*, 2018), authors proposed binary classification model comprised of linear SVM as the model training block. For detection, features like bag of words, character and word n-grams, lexicon-based features and few others were used. The F1 scores for the best combinations for English and Dutch language was 0.64 and 0.58 signifying a lot of room for improvement. In one such similar studies, authors used Naïve Bayes classifier as the base model along with few features for cyberbullying detection. The model yielded considerably better precision score of about 0.77. (Anggraini, Sucipto and Indriati, 2018)

In one study by (Zhang *et al.*, 2019), authors used slightly advance detection approaches like Word2Vec, N-gram and Doc2Vec for detection of cyberbullying on Japanese text on twitter. For model generation authors used a host of models were used ranging from logistic regression to deep learning models. One interesting insight that the study revealed that character N-grams are very efficient among all the detection approaches used. One such methods performed similar experiments using bag-of-words method for multilingual detection (Pawar and Raje, 2019).

Many approaches focused on implementing rule-based detection technique along with machine learning techniques for cyberbullying detection. The rules ranged from network based rules to language profanity based rules for detecting the cyberbullying instances (Nurrahmi and Nurjanah, 2018) (Reynolds, Kontostathis and Edwards, 2011).

2.3 Hybrid detection techniques

This subsection highlights all the hybrid detection techniques presented by the works produced in the past for the task of cyberbullying detection.

(Ptaszynski *et al.*, 2019) proposed a novel system of using brute force search method along with combination of other pattern extraction technique in order to detect malicious and abusive content. The method was quite impressive then the traditional feature-based detection but computationally intensive and time consuming. In one of the studies, the authors combined social, syntactic, sentiment and semantic features for the detection. For every sentence, input vector was in relation with weights of assigned features.

Further in one of the studies, the authors proposed a socio-linguistic method for identifying the interrelations between the user and the participant's social interaction. For this, the authors used n-grams++, latent linguistic, socio linguistic, seeds++ and n-grams along with Probabilistic soft logic (PSL) to perform the classification.

2.4 Advance detection techniques

A lot of studies came up with novel detection methods in order to improve the efficiency of detection. This sub-section aims at investigating these methods in brief.

In one of the papers, (Shekhar, 2018) proposed a novel bag-of-phonetic-codes method that surpassed the traditional bag-of-words method in terms of detection efficiency. It took the pronunciation of words into account in order to detect the censored and misspelled words that are often a part of social media texting. The model achieved a significant accuracy of 97%.

Another such study (Zhang *et al.*, 2017) integrated pronunciation of words as a feature but used deep learning for training the model instead of machine learning as opposed to previous approach by (Shekhar, 2018).

(Nurrahmi and Nurjanah, 2018) proposed a system employing a novel technique fuzzy fingerprint in order to detect the cyberbullying incidents. The technique focused on placing similar words in the same block just like the concept of word embedding. One of the advantages that was seen in the method was that it could perform quite well as compared to the traditional detection approaches.

2.5 Word Embedding detection

This subsection aims at investigating the latest advancement in the NLP i.e. word embedding detection for the detection of cyberbullying.

In one of the studies by (Zhao, Zhou and Mao, 2016), the authors proposed a novel method Enhanced Bag of words method inspired by word embeddings in order to carry out the detection. The underlying method used in this system was word2vec embedding for the purpose of encoding relationship between the words into vectors. The authors performed an extensive comparison of the proposed method with other traditional detection methods like BOW, EBOW, SBOW, LDA and LSA. The proposed model outperformed the methods across all the metrics with a precision score of 76.8%, recall of 79.4% and F1 score of 78%. Although the method looked great on the surface level but since this method is based on word2vec, it often lacks the ability deal with out of vocabulary words. Such kind of method would work fine on large datasets but would struggle when the size of the dataset is comparatively small.

In another study by (Al-Ajlan and Ykhlef, 2018), the authors used have incorporated GloVe word embedding along with Convolution Neural Network on twitter dataset for detection of cyberbullying. The study presented thorough comparison of the applied model with one of the most widely used method for cyberbullying detection that is SVM. The model significantly outperformed the counterpart with an overall accuracy of 95% as compared to the SVM's accuracy of 81% indicating an overall increase of 12% in accuracy.

In another such study by (Agrawal and Awekar, 2018), proposed a novel system that integrated deep neural network (DNN) models with word embedding methods in order to detect cyberbullying. DNN models used for the study were CNN, BLSTM, LSTM and BLSTM with attention. The authors also applied cross domain testing that is transfer learning in order to test the consistency and robustness of models. Word embedding methods used in the study was GloVe and SSWE. GloVe embedding has been a popular choice in the NLP but often it fails to understand the semantic meaning conveyed in the sentence. Also, the models performed quite well in the same domain testing but performance was affected a lot on cross domain testing. Inspired by the work done by (Agrawal and Awekar, 2018), this study will focus on addressing this issue and try to improve the detection efficiency on complete transfer learning using the newly introduced word embedding methods

In conclusion, it could be seen that a lot of research has been done in the detection of cyberbullying and the literature review a lot of interesting approaches to carry out the task of

detection. But it could be seen that there are a lot of issues are still unresolved. This study aims at addressing these bottlenecks and further the detection efficiency of cyberbullying detection.

3 Research Methodology

The methodology followed for the purpose of this study was Knowledge Discovery Databases (KDD) methodology. All the stages prescribed in this methodology were systematically followed in order to extract meaningful insights from the data.

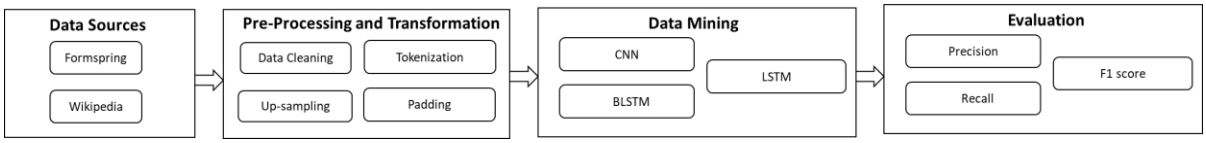

Figure 1: Research Methodology

3.1 Dataset Description

The datasets used for the purpose of the study were requested from (Agrawal and Awekar, 2018). Two datasets namely Formspring (Q&A platform) and Wikipedia talk page data (Collaborative knowledge platform) were used in this research study. Both of the datasets were manually labelled and were available for public use. The details regarding each of the dataset is as follows:

Formspring data: Formspring was an anonymous question and answer based social network. The dataset consists a total of 12,000 pairs of questions and answers that were annotated by three experts. Out of 12k pairs, 825 pairs were labelled as cyberbullying upon the mutual agreement of at least two experts.

Wikipedia talk page data: This dataset consist of over 100k comments from the Wikipedia talk pages. The comments consist of the discussion among the editors that have contributed to the page. 13,590 comments were manually labelled as personal attack among the mutual consent by 10 experts.

3.2 Data Pre-Processing

Data pre-processing is one of the most important steps in implementation of any data analytics project. It is very necessary to ensure that the data is clean without any noise that may affect the credibility of the result. Following steps were taken as a part of pre-processing for the system.

1. First of all, it was checked that the data consist of any null values.

2. One of the main assumptions that we need to satisfy before implementing the model is that the data is balanced. A key thing that was observed in the two datasets is both of the datasets were very skewed with high amount of non-cyberbullying instances as compared to cyberbullying instances. As a result, the minority class in both the datasets were up sampled to match the proportion of majority class. The distribution of counts of before and after oversampling for Formspring dataset can be seen in figure 2.

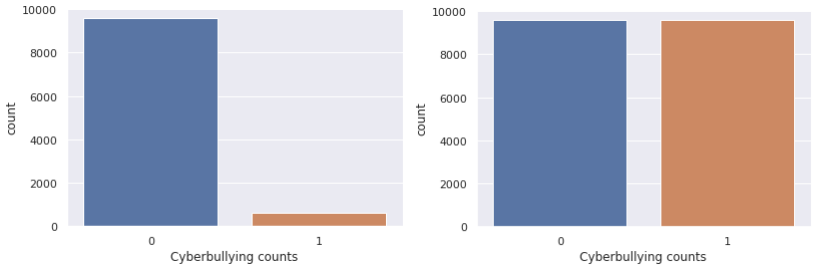

Figure 2: Before vs After Oversampling (Formspring data)

3. The textual data was converted to string and the label column was converted to categories. Later the label column was converted to dummies before feeding it to the DNN models.

4. The text data was then subject to tokenizing in order to create tokens for each word encountered in the dataset.

5. The word count in each of the posts greatly varies across the datasets. This could affect the amount of distinct words encountered in the datasets. Hence to create a standard post length, all the long posts were truncated to a length of 300 words in both of the datasets.

6. fastText and ELMo embeddings are computationally very demanding and face memory constraints when training on large datasets. In this study, Wikipedia dataset is extremely large with more than 100k rows. In order to deal with this problem, the Wikipedia data sliced and reduced before feeding it to the word embedding block.

3.3 Transformation

Since word embedding has been integrated in the model, it was necessary to first tokenize the words present in the training data and create a vector space of vocabulary. This vector space of vocabulary is fed to the DNN models as an input. After that, an empty matrix consisting of zeros with the row length equal to vocabulary length and the column length equal to the embedding dimension was instantiated. Once the empty matrix is created, the words in the vocabulary from the datasets are then mapped to the dictionary of words present in the pretrained vectors. This particular step maps each and every word in the sentence in a vector space where words with semantically similar meanings are located closed to each other. The embedding matrix at the end of this step is then applied as weights in the DNN models which

perform the further classification. While performing the testing, the testing data is also tokenized and converted to vector space before applying it to the prediction model.

3.4 Data mining

Data mining is the step where hidden information and patterns are extracted from the data. Once the pre processing and feature engineering is completed, the transformed data and the embedding weights are applied to the Deep Neural Network (DNN) models that further perform the classification. DNN models have proven to outperform typical machine learning models in the task of classification. Three popular DNN models are used for the purpose of this research namely LSTM, CNN and BLSTM.

3.5 Evaluation

The dataset is first divided into training and testing data using the train-test split method in the 80:20 ratio. For evaluation of the models, same evaluation parameters are used in the study by (Agrawal and Awekar, 2018) in order to standardize the comparison. Various combination of DNN models and word embedding methods are rigorously evaluated on the basis of three main evaluation metrics i.e. precision, recall and F1 score. Further to test the robustness of the models, transfer learning is applied where a model trained on a particular dataset (e.g. Formspring) is tested on a completely new dataset (e.g. Wikipedia).

4 Design Specification and Implementation

4.1 System Design

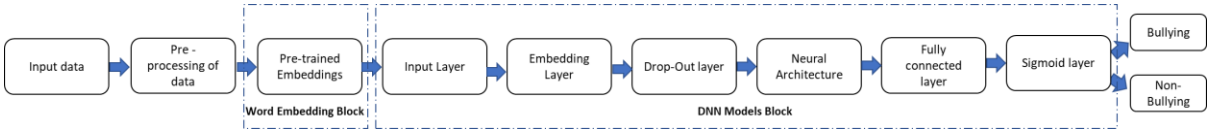

Figure 3: Model Architecture

The entire system basically consists of two main components:

1. Word Embedding Block
2. DNN Block

The following subsection will discuss the two components in detail.

Word Embedding Block

Word embeddings is a technique in which textual data is converted to numerical representation. The words with semantically similar meaning are located close to each other in the vector space on the basis of cosine similarity between them. As you can see from figure

3, semantically similar words with high cosine similarity are grouped together in the vector space.

Similar Words	Cosine Similarity Scores
a slut	0.815
whore	0.738
a whore	0.638
hypocrite	0.536
bitch	0.508
puta	0.468
nerd	0.455
bully her	0.451
fat bully	0.440
bully nigga	0.435

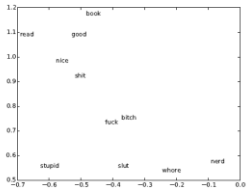

Figure 4: Example of similar words in vector space
(Zhao, Zhou and Mao, 2016)

The word embedding methods used in this research are fastText, ELMo and stacked embedding (flair-forward + flair-backward + GloVe). fastText embedding works on the principle of n-gram representation. It creates multiple character n-grams for each word in the vocabulary. This helps fastText to learn vector representations for out of vocabulary words not in the training corpus. [2]

ELMo and stacked embedding on the other hand are language models and work on the principle of assigning tokens as a function of specific sentence. This helps these models understand the context behind the sentence.

1. I will go to 'bank' tomorrow.
2. I have kept power 'bank' for charging.

In the above examples, a typical word embedding approach like GloVe, word2vec, SSWE, etc will fail to differentiate between the two sentences. But since ELMo and flair embeddings generate tokens on sentence level, that makes them more context aware and can help them understand the difference between the two sentences. This can prove as a major breakthrough in sentiment analysis-based task like cyberbullying.[3]

To integrate word embeddings in the methodology, the sentences in the data were first tokenized before feeding into word embedding block. (Tokenization is nothing but breaking down each sentence into distinct tokens i.e. words). The tokens were then mapped with the repository of pre-defined words of each word embedding method and the corresponding vectors were assigned to each word. The end product of the embedding block is the embedding matrix

[2] https://medium.com/@adityamohanty/understanding-fasttext-an-embedding-to-look-forward-to-3ee9aa08787
[3] https://www.analyticsvidhya.com/blog/2019/02/flair-nlp-library-python/

consisting of the vectors (numerical representation) of the vocabulary in the dataset. This embedding matrix is further passed to one of the deep learning models (CNN/ BLSTM/LSTM) selected for training.

General set up requirement of word embeddings: -

i. Creation of vector space by tokenizing the posts and serializing the tokens.
ii. Padding the sequences as per the input length specified where,
input length: Length of each vector i.e. maximum number of words allowed in post
iii. Creation and Initialization of embedding matrix. The dimension of the matrix is decided on the basis of two parameters - Length of vocabulary and Embedding dimension of the word embedding method where,
Length of vocabulary: Total amount of words in the vectors after tokenization.
and
Embedding dimension: Default value of dimension of word embedding method used.
iv. Mapping of embedding - This step involves mapping of each word in the vocabulary of data to the vector value defined in the pre-trained embedding.
v. The tokenized vector space is served as input to the DNN models and the embedding matrix is served as embedding weights to the models.

DNN Block

The second highlighted part in the part of the design as seen in figure 2 is the DNN models block. Inspired by the work done by (Agrawal and Awekar, 2018), an attempt was made to replicate the three models that were used in the study that are CNN, LSTM and BLSTM. The models are very similar in architecture with minimal difference in the neural architecture. The models have been mentioned in the increasing order of complexity. Convolution Neural Network (CNN) that have been traditionally used in image processing have proven quite impressive in natural language processing. CNN have demonstrated exceptional capabilities in sentiment classification tasks. A good example of use of CNN for sentiment analysis can be observed in (Kim, 2014). Long Short-Term Memory (LSTM) model is quite good at learning long term dependencies. LSTM is a special kind of Recurrent Neural Network (RNN) have feedback connections that help the model to process entire sequences of data. The presence of internal memory LSTM gives it the capability to process random sequences of data that gives it an edge in textual classification of data. (Johnson and Zhang, 2016) Bidirectional LSTMs have the inherent capability of encoding the data in both forward and backward direction that helps them process more input information from the textual data. (Zhou *et al.*, 2016) The embedding layer is common among all three models and is one of the most important layers in the architecture. The embedding matrix created by mapping the vectors in the word embedding block is given as an input to this layer. This layer has the ability to improve upon the initial embeddings provided by the word embedding block and learn task specific embedding that helps the model understand the context of the sentence and classify the cyberbullying instances. One of the problems that the deep learning models often face is overfitting that often create a mirage of unrealistic results that the model is not capable of. To avoid such problem, drop out layers have been introduced wherever required in order

to prevent overfitting. Fully connected layer is nothing but dense layer with the number of neurons same as the number of classes in the target variable. Final layer is the SoftMax/Sigmoid layer that perform the final classification. All three models used in this study are trained using backpropagation. The loss function present in the models is categorical cross-entropy and adam optimizer is used as the optimizer in all the three models.

4.2 System Implementation

The section will discuss the end to end implementation that was undertaken for the purpose of this study. First of all, the entire implementation of the study was conducted on an open source cloud platform Google Collaboratory. The reason for using Google Colab instead of Jupyter Notebook or any other integrated development environment is because of its smooth integration with the google drive and provision of GPU in case of heavy processing-oriented tasks. The language used for the purpose of programming is Python because of its huge support for the deep learning, NLP and word embedding libraries and relative ease of use. All of the supporting files were uploaded to google drive and the drive was mounted to the drive to let the colab have access to the files.

Two datasets were used for the purpose of this study that is formspring data and wiki talk page data. Both the datasets consist of an input variable containing the textual data from the post and a target variable containing the labels annotated by experts that the post contains cyberbullying or not. The data was first imported from the drive and stored in a dataframe. After importing, the data was checked if there are any missing values. In contrary to the process flow followed by (Agrawal and Awekar, 2018) where the train-test split is performed after the oversampling, we have performed the train-test split before oversampling. The oversampling is only done on the train data and not the test data. The reason for this is sampling the data before the train-test split results in overfitting of data and may give overestimated representation of the performance of the model when the model is evaluated. Further value counts of each class were inspected to see how balanced the data is. On inspection it was noted that the data was very skewed to the non-cyberbullying class. As a result, up sampling was performed on the data to balance the minority class that is cyberbullying instances to majority class with non-bullying instances.

Once the pre-processing is done, the data is transferred to the word embedding block where the data is transformed to vectorized format before feeding into the models. The word embedding block also creates an embedding matrix by mapping the words with the pretrained vectors. Detailed explanation of the word embedding block can be seen in the (sub-section 4.1). Further the input vectors and the embedding matrix is passed to the DNN models block where model is trained. DNN models used for the purpose of the study are CNN, LSTM and BLSTM. Detailed explanation of the architecture of the models can be traced back to subsection 4.1.

The test data is further used to perform the predictions using the trained model. Various combination of models and word embedding method were tried and tested. Further the performance of the models was evaluated using the evaluation metrics precision, recall and

F1 score. The evaluation scores and the performance of the models can be seen in the upcoming section 6. To further test the robustness of the models, transfer learning where the model trained on one particular dataset is tested on another dataset. Finally, the performance of the model is evaluated and compared on normal testing and cross platform testing that is transfer learning.

5 Evaluation

The primary goal of the research was to investigate if the newly introduced word embedding methods fastText, ELMo and flair stacked embedding are able to further the detection efficiency of cyberbullying detection. To get the real sense of idea of the performance of this word embedding methods, an attempt was made to replicate the system design of (Agrawal and Awekar, 2018) as much as possible as mentioned in section 4.1 in details.

For the setup, three DNN models viz. CNN, LSTM and BLSTM are used for the purpose of training the data. For the detection block, three embedding methods fastText, ELMo and flair stacked embedding are used. First experiment i.e. sub-section 5.1 aims at using fastText embedding with different combination of DNN models like CNN, LSTM and BLSTM. Second experiment i.e. sub-section 5.2 aims at using the best model obtained in experiment 1 along with different word embedding methods like fastText, ELMo and stacked flair embedding. Finally, after training the model, transfer learning is applied on the models in experiment 3 i.e. sub-section 5.3 where the models are subjected to cross domain testing on a completely new dataset. The results obtained in sub-section 5.3 were used to compare the results of complete transfer learning obtained in the study (Agrawal and Awekar, 2018) where BLSTM with attention model was used.

5.1 Experiment 1: Using fast text embedding with different deep learning models

The primary goal of this experiment to evaluate which DNN model gave the best results in training the initial word embeddings.

Dataset	Label	Precision			Recall			F1 Score		
		BLSTM	CNN	LSTM	BLSTM	CNN	LSTM	BLSTM	CNN	LSTM
F+	Bully	0.89	0.90	0.90	0.69	0.75	0.66	0.78	0.82	0.75
W+	Attack	0.78	0.80	0.79	0.47	0.56	0.50	0.56	0.64	0.58

Table 1: Performance comparison of DNN models keeping fastText embedding constant

For the purpose of this experiment, fastText word embedding was chosen as the constant across the evaluation of models. Table 1 describes the performance of three DNN models in

two datasets Formspring (F+) and Wikipedia (W+), keeping the word embedding method fastText constant.

As it can be clearly seen in table 1, CNN model performed the best across both of the datasets with highest F1 score of 0.82 in case of formspring dataset and F1 score of 0.64 in Wikipedia dataset. LSTM and BLSTM yielded similar scores in Formspring and Wikipedia dataset. BLSTM performed better in case of the Formspring dataset with an F1 score of 0.78 against LSTM's score of 0.75. While LSTM performed better in case of Wikipedia dataset with an F1 score of 0.58 as compared to the score of 0.56 of BLSTM. Overall CNN performed the best in both datasets with a precision, recall and F1 score of 0.90, 0.75 and 0.82 respectively on formspring data and 0.80, 0.56 and 0.64 on Wikipedia data.

This experiment aimed at investigating which DNN model yielded the best performance in classifying the cyberbullying instances. The best performing model CNN will be applied to the following experiment 2 in order to evaluate the performances of different word embedding methods fastText, ELMo and flair stacked embedding.

5.2 Experiment 2: Using the best performing model CNN with various embedding methods

This experiment aimed at identifying the best performing word embedding approach while keeping the best model from the experiment 1 i.e. CNN constant.

Dataset	Label	Precision			Recall			F1 - Score		
		fastText	ELMo	Flair with Glove	fastText	ELMo	Flair with Glove	fastText	ELMo	Flair with Glove
F+	Bully	0.90	0.89	0.90	0.75	0.76	0.75	0.82	0.82	0.81
W+	Attack	0.80	0.84	0.84	0.56	0.81	0.80	0.64	0.82	0.82

Table 2: Performance of various word embedding methods keeping training model CNN constant

The above table represents the evaluation of different word embedding methods fastText, ELMo and flair stacked embedding (Flair + GloVe) across two datasets Formspring (F+) and Wikipedia (W+) keeping one DNN model i.e. CNN constant

As it can be seen in table 1, all of the detection approaches tried with CNN yielded very similar results with ELMo having a slight edge over the others. For the formspring data, fastText and ELMo embedding yielded same results in terms of F1 score that is a score of 0.82. In case of Wikipedia data, ELMo and stacked embedding (Flair + glove) yielded slightly better results than fastText. F1 score of ELMo and stacked embedding was 0.82 while the fastText yielded an F1 score of 0.64.

This experiment aimed at finding the best word embedding approach that can be applied for transfer learning to be carried out in experiment 3.

5.3 Experiment 3: Performing cross platform testing (transfer learning) to test the robustness of the models.

Post the training of the models with the word embedding methods, the models were subjected to cross domain testing in order to test the robustness of the models when tested on a completely new dataset. Table 3 also includes a comparison of scores of complete transfer

learning of the base research (Agrawal and Awekar, 2018) and the results obtained in this research.

Metric	Train / Test	Research	F+	W+
Precision	F	Old	-	0.51
		New		0.89
	W	Old	0.82	-
		New	0.79	
Recall	F	Old	-	0.66
		New		0.47
	W	Old	0.21	-
		New	0.68	
F1 score	F	Old	-	0.58
		New		0.59
	W	Old	0.35	-
		New	0.73	

Table 3: Comparison of previous paper vs this study

Table 3 illustrates the transfer learning applied on the two datasets and comparison of scores to the results obtained by the authors in the study(Agrawal and Awekar, 2018) . The dataset used for training are denoted by F+ and W+ and the dataset used for testing are denoted by F and W. Since the experiment is primarily aimed at evaluating transfer learning on the datasets, the scores of the models tested on the same dataset are not mentioned in this table and denoted by a '-'.

From the results illustrated in the above table, it can be clearly seen that the best performing combination of this study (CNN model + fastText embedding) outperforms the results obtained in the existing research. The comparison can be broadly divided in two transfer learning experiments performed in this study.

1. **Train data - Formspring and Test data - Wikipedia**

In this section of the experiment, the model was trained on Formspring data and tested on Wikipedia data. There was not much difference in the precision of the model as compared to the previous model with a slight decline of 0.3 points in this study. The model outperformed majorly in terms of recall as compared to the results obtained by (Agrawal and Awekar, 2018) with an increase from 0.21 to 0.68. The overall F1 score of the new model increased to 0.73 from 0.35. All in all, it was observed that that the new model significantly outperformed the model used by (Agrawal and Awekar, 2018)

2. Train data - Wikipedia and Test data - Formspring

This sub section of the experiment focused at training the model on Wikipedia data and testing it on Formspring data. From the table above, it can be seen that there is a significant increase from 0.51 to 0.89 in the precision of the model as compared to the previous study. There was a slight decline of 0.66 to 0.47 in the recall of the model. The F1 score of the model was in parity with the score that previous study had obtained with an increase from 0.58 to 0.59. It can be concluded that the model performed similarly to the base study when trained on Wikipedia and tested on formspring.

5.4 Discussion

The results were evaluated on the basis of same evaluation metrics used by (Agrawal and Awekar, 2018) for a systematic and standardized comparison. As seen in section 5.1, CNN is the best performing model across all scenarios in terms of DNN models. Among the word embedding methods used, a speed - performance trade off was observed. As seen in section 5.2, ELMo and stacked embedding yielded slightly superior performance to that of fastText but they were computationally intensive and increased the model training time significantly. On the other hand, fastText was comparatively faster and computationally less demanding. ELMo and stacked embedding are an ideal choice when the size of the dataset is comparatively small. But in case of large datasets, fastText embedding would be a more sensible option.

The study was able to achieve better scores in transfer learning as compared to the baseline paper as observed in experiment 3 i.e. sub-section 5.3. Two things were done differently in this paper as compared to the work done by (Agrawal and Awekar, 2018). Firstly, to deal with the class imbalance, slightly different approach for sampling was used. (Arango, Pérez and Poblete, 2020) in their study reviewed the experiments performed by (Agrawal and Awekar, 2018) where they highlighted the issue of performing the train-test split after sampling of data which resulted in overfitting and results were significantly overestimated. This study took that into account, train-test split has been done before sampling the data as explained in the system implementation section 4.2 in detail. Thus, the problem of overfitting was avoided in this study. Secondly, the word embedding approaches used in this paper i.e. fastText, ELMo and stacked flair embedding are more context aware and proficient in dealing with out of vocabulary words as explained in section 4.1 in detail. That helped in improving the detection efficiency even further.

Improved detection efficiency means enhancement in the capacity of the models to distinguish between cyberbullying and non-cyberbullying instances more precisely and accurately with reduced misclassifications. Better scores in the third experiment (section 5.3) i.e. transfer learning indicate that models perform well even if they are subjected to a completely new dataset without any additional training. Such kind of models are more robust to variation in the language and bullying style when implemented in real world dynamic environment.

6 Conclusion and Future Work

The major purpose of the research study was to integrate newly introduced word embedding methods fastText, ELMo and stacked flair embedding in the cyberbullying detection

workflow and investigate if it can further the cyberbullying detection efficiency. The word embedding approaches used in the study were able to further the detection efficiency of the models thereby doing justice to the underlying research question of the study.

The research was successful in investigating the challenges and limitations in the state-of-the-art and overcome the limitations faced in the studies. One of the most major objectives of the study was to develop the models that could perform well on a completely new platform without any additional training. The final model CNN with fastText embedding was subjected to transfer learning after the initial training. The model when subjected to transfer learning performed significantly well with satisfactory F1 score of 0.73 on Formspring data and F1 score of 0.59 on Wikipedia data. Out of all the DNN models, the model that performed the best across all the scenarios was CNN as seen in section 5.1. ELMo was the best performing word embedding method while fastText was computationally faster than the other two embedding methods as seen in section 5.2.

One of the key limitations of the study was that even if stacked flair embedding and ELMo performed significantly well in terms of evaluation metrics, they were significantly slow and had high GPU requirements. As a result, while training the model on large dataset like Wikipedia, the model failed to allocate tensors when the resources were exhausted. Hence to avoid that problem, Wikipedia data was sliced before feeding it to the word embedding block.

The future work of this study would focus on integrating more features about the profile information and social graph of the users involved in the cyberbullying in order to get deeper insights and understand the context behind the cyberbullying incidents. Moreover, the study may focus on reducing the computational requirement of the models so that it can take on large datasets without any modifications.

Acknowledgement

I would like to express my sincere gratitude towards my supervisors Prof. Paul Stynes and Prof. Pramod Pathak for continual support and guidance throughout the implementation of the research project.

References

Agrawal, S. and Awekar, A. (2018) 'Deep learning for detecting cyberbullying across multiple social media platforms', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10772 LNCS(Table 2), pp. 141–153. doi: 10.1007/978-3-319-76941-7_11.

Akbik, A., Blythe, D. and Vollgraf, R. (2018) 'Contextual String Embeddings for Sequence Labeling', *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649. Available at: https://github.com/zalandoresearch/flair.

Al-Ajlan, M. A. and Ykhlef, M. (2018) 'Deep learning algorithm for cyberbullying detection', *International Journal of Advanced Computer Science and Applications*, 9(9), pp. 199–205. doi: 10.14569/ijacsa.2018.090927.

Anggraini, I. Y., Sucipto, S. and Indriati, R. (2018) 'Cyberbullying Detection Modelling at Twitter Social Networking', *JUITA : Jurnal Informatika*, 6(2), p. 113. doi: 10.30595/juita.v6i2.3350.

Arango, A., Pérez, J. and Poblete, B. (2020) 'Hate speech detection is not as easy as you may think: A closer look at model validation (extended version)', *Information Systems*. doi: 10.1016/j.is.2020.101584.

Bojanowski, P. *et al.* (2017) 'Enriching Word Vectors with Subword Information', *Transactions of the Association for Computational Linguistics*, 5, pp. 135–146. doi: 10.1162/tacl_a_00051.

Van Hee, C. *et al.* (2018) 'Automatic detection of cyberbullying in social media text', *PLoS ONE*, 13(10), pp. 1–21. doi: 10.1371/journal.pone.0203794.

Johnson, R. and Zhang, T. (2016) 'Supervised and semi-supervised text categorization using LSTM for region embeddings', *33rd International Conference on Machine Learning, ICML 2016*, 2, pp. 794–802.

Kim, Y. (2014) 'Convolutional neural networks for sentence classification', *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1746–1751. doi: 10.3115/v1/d14-1181.

Nurrahmi, H. and Nurjanah, D. (2018) 'Indonesian Twitter Cyberbullying Detection using Text Classification and User Credibility', *2018 International Conference on Information and Communications Technology, ICOIACT 2018*. IEEE, 2018-Janua, pp. 543–548. doi: 10.1109/ICOIACT.2018.8350758.

Pawar, R. and Raje, R. R. (2019) 'Multilingual cyberbullying detection system', *IEEE International Conference on Electro Information Technology*. IEEE, 2019-May, pp. 040–044. doi: 10.1109/EIT.2019.8833846.

Ptaszynski, M. *et al.* (2019) 'Brute-force sentence pattern extortion from harmful messages for cyberbullying detection', *Journal of the Association for Information Systems*, 20(8), pp. 1075–1127. doi: 10.17705/1jais.00562.

Reynolds, K., Kontostathis, A. and Edwards, L. (2011) 'Using machine learning to detect cyberbullying', *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011*. IEEE, 2, pp. 241–244. doi: 10.1109/ICMLA.2011.152.

Romsaiyud, W. *et al.* (2017) 'Automated cyberbullying detection using clustering appearance patterns', *2017 9th International Conference on Knowledge and Smart Technology: Crunching Information of Everything, KST 2017*. IEEE, pp. 242–247. doi: 10.1109/KST.2017.7886127.

Shekhar, A. (2018) 'A Bag-of-Phonetic-Codes Model for Cyber- Bullying Detection in Twitter', *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*. IEEE, pp. 1–7.

Singh, V. K., Huang, Q. and Atrey, P. K. (2016) 'Cyberbullying detection using probabilistic socio-textual information fusion', *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*. IEEE, pp. 884–887. doi: 10.1109/ASONAM.2016.7752342.

Zhang, J. *et al.* (2019) ‘Cyberbullying Detection on Twitter using Multiple Textual Features’, *2019 IEEE 10th International Conference on Awareness Science and Technology, iCAST 2019 - Proceedings*. IEEE, pp. 1–6. doi: 10.1109/ICAwST.2019.8923186.

Zhang, X. *et al.* (2017) ‘Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network’, *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 740–745. doi: 10.1109/icmla.2016.0132.

Zhao, R., Zhou, A. and Mao, K. (2016) ‘Automatic detection of cyberbullying on social networks based on bullying features’, *ACM International Conference Proceeding Series*, 04-07-Janu(January). doi: 10.1145/2833312.2849567.

Zhou, P. *et al.* (2016) ‘Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling’, *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*, 2(1), pp. 3485–3495.

Zois, D. S. *et al.* (2018) ‘Optimal Online Cyberbullying Detection’, *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. IEEE, 2018-April, pp. 2017–2021. doi: 10.1109/ICASSP.2018.8462092.