PERFORMANCE EVALUATION FOR THE USE OF ELMO WORD EMBEDDING IN CYBERBULLYING DETECTION

Tina Yazdizadeh and Wei Shi

School of Information Technology,
Carleton University, Ottawa, Ontario, Canada

ABSTRACT

Communication using modern internet technologies has revolutionized the ways humans exchange information.. Despite the numerous advantages offered by such technology, its applicability is still limited due to problems stemming from personal attacks and pseudo-attacks. On social media platforms, these toxic contents may take the form of texts (e.g., online chats, emails), speech, and even images and movie clips. Because the cyberbullying of an individual via the use of such toxic digital content may have severe consequences, it is essential to design and implement, among others, various techniques to automatically detect, using machine learning approaches, cyberbullying on social media. It is important to use word embedding techniques to represent words for text analysis, typically in the form of a real-valued vector that encodes the meaning of words. The extracted embeddings are used to decide if a digital input contains cyberbullying contents. Supplying strong word representations to classification methods is a key facet of such detection approaches. In this paper, we evaluate the ELMo word embedding against three other word embeddings, namely, TF-IDF, Word2Vec, and BERT, using three basic machine learning models and four deep learning models. The results show that the ELMo word embeddings have the best results when combined with neural network-based machine learning models.

KEYWORDS

Cyberbullying, Natural Language Processing, Word Embeddings, ELMo, Machine Learning.

1. INTRODUCTION

Cyberbullying is a real-life issue that comes from the development and global use of Information and Communication Technology (ICT) solutions in today's life. It endangers everyone's life, especially children, meaning the future psychological health of societies is at real risk. Cyberbullying detection owes its development to many Artificial Intelligence (AI)-based methods. This means a set of semantic and sentiment analysis through data pre-processing, word embeddings, and classification is performed to make sure that the toxic text-based concepts are accurately detected.

The Advancement of ICT has led to the explosion of online communication via social networks and other related applications. Communication enabled by internet technologies has revolutionized modern human interaction. People would like to connect to each other over social media for many reasons, including expressing their ideas and opinions, engaging in forums and discussions, and receiving feedback on their views via interactive media. Despite all the advantages made available by ICT, its applicability is limited due to the problems caused by

David C. Wyld et al. (Eds): ARIA, SIPR, SOFEA, CSEN, DSML, NLP, EDTECH, NCWC - 2022
pp. 131-144, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121511

personal attacks or pseudo-attacks through the usage of toxic content. Therefore, it is crucial to design and implement various techniques to detect cyberbullying content on social media automatically and evaluate the effectiveness of various approaches.

The Semantic and Sentiment Analysis (SSA) technique[1] is frequently used for cyberbullying detection in texts. In semantic analysis, the meaning of a given text is drawn using computer programs that interpret sentences, paragraphs, or whole documents, by analyzing the grammatical structure and identifying relationships between individual words in a particular context. On the other hand, sentiment analysis employs Natural Language Processing (NLP) techniques, text analysis methods, and in general computational linguistics to systematically identify, extract, quantify, and study affective states and subjective information as what needs to be done to identify cyberbullying contents. Both of these two techniques usually employ supervised Machine Learning (ML) techniques to perform cyberbullying detections. It is essential to use rich datasets to perform training in Neural Networks (NN) and Deep Learning (DL) based solutions.

Word embedding techniques are used to represent the words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that they are closer in the vector space, expected to be similar in meaning. Word embedding paves the way for representing textual data ready to be fed to the ML tools for further analysis toward cyberbullying detection. It is a mapping from the words space with different dimensions to real numbers space with much lower dimensions. Word to Vector (Word2Vec) word embedding model was designed and presented in 2013 by researchers from Google[2].Bidirectional Encoder Representations from Transformers (BERT)[3] were also proposed by a Google team in 2018.

Embeddings from Language Model (ELMo) was first introduced by Matthew E. Peters et al. as a new type of deep contextualized word representation that models both complex characteristics of words and the procedure through which these vary across linguistic contexts[4]. ELMo can analyze the syntax and semantics of the texts in a very prominent manner. It captures semantic relationships as well as syntactic relationships. That is why it achieves good results in solving the problem of polysemous words and outperform previously existing word embeddings. ELMo has been known as a very effective method for word embedding in many applications. In this paper, we employ ELMo as a word embedding technique that, in conjunction with deep learning models and MLP classifier, has provided us with a novel structure to perform cyberbullying detection on well-known datasets. The proposed structure benefits from the most important and influential tools for word embedding and classification that paves the way for more accurate results. The contributions of this paper are summarized as follows:

(i) We combine ELMo with Multi-Layer Perceptron (MLP), Decision Tree, and Random Forest to achieve text-based cyberbullying detection. The combination of ELMo with MLP provided us with better results in terms of precision, recall, and F1-score in comparison to the previous research works using MLP with TF-IDF word embedding. To the best of our knowledge, the combination of ELMo with the Decision Tree has not been used previously.
(ii) We conduct a comparative evaluation of the impact of ELMo word embedding on three basic machine learning models and four deep learning models. Six different datasets were used to evaluate the performance of the models using three metrics. Results demonstrate the advantage of ELMo on cyberbullying detection when combined with neural network-based machine learning models.
(iii) Among the deep learning models, we combine ELMo with a modified Dense model that leads to further improvement compared to previous research works.

2. LITERATURE REVIEW AND BACKGROUND

In the past years, researchers have done several works on NLP and text analysis in social media for cyberbullying detection. They used a wide variety of Machine Learning (ML) algorithms such as Support Vector Machine (SVM), Ensemble Models, Linear Regression, and Naive Bayes by using Deep Learning (DL) models on different datasets such as Twitter, Facebook, FormSpring, and so on. In this section, we review the most recent and reputable references in the field.

Deep Learning (DL) technique has been used by the authors of [1] and [5]. The main goal of the papers is to ease online communication on textual platforms without being hurt by insults, harassment, and fake news. This is one step forward toward fully AI-based techniques for the detection and prevention toward the protection of a reader being hurt during online chatting. As a general drawback, the computational burden in DL-based techniques is a matter to be addressed. Bidirectional Encoder Representations from Transformers (BERT)[3], as a deep bidirectional, unsupervised language representation capable of creating word embedding (that represents the semantic of the words in the context that they are used) along with other methods is also used in this paper. The four employed deep learning models are Dense, Convolutional Neural Network (CNN), and Long-Short Term Memory (LSTM) layers to detect various levels of toxicity. As for word embedding techniques, the paper has examined Word2Vec[2] and BERT[3] algorithms. To show the performance of the proposed method, the authors have employed the dataset that was released by a Kaggle competition [6] collected from Wikipedia comments, which have been manually labeled into six different toxicity classes.

In another recently published survey paper, the authors have reviewed related works in the literature where word embeddings techniques based on deep learning techniques have been used[7]. Moreover, different types of word embeddings are categorized in this paper. These models need to understand how to pick out keywords that can change the emotion of a sentence. The popular models with the capability of solving such cases are ELMo, OpenAI-GPT, and BERT.

More related to the application discussed in this paper, the effectiveness of the pre-trained embedding model using deep learning methods for classification of emails is examined in [8]. Global Vectors (GloVe) and BERT pre-trained word embedding are employed to identify relationships between words for the categorization of the emails. Well-known datasets like Spam Assassin and Enron are used in the experimentation. In the evaluation phase, the confusion matrix, accuracy, precision, recall, F1-score, and execution time with 10-fold cross-validation are computed for each method. The results show that the CNN model with GloVe embedding gives slightly better accuracy than the model with BERT embedding and traditional machine learning algorithms.

A survey on embeddings in Clinical Natural Language Processing has been given in[9]. Various medical corpora and their characteristics and medical codes have been discussed in this paper. The paper also explores that ELMo generates context-dependent vector representations and hence accounts for the polysemy nature of word embeddings for Out of Vocabulary (OOV), misspelled, and rare words. The main disadvantage of ELMo is computationally intensive, and memory requirements increase with the size of the corpus. ELMo is different from other well-known embedding techniques as it makes use of all the three-layer vectors, i.e., the final representation of a word is obtained as a task-specific weighted average of all the three-layer vectors. ELMo vectors are deep because they come through three-layer vectors and are context-sensitive because they assign different representations to a word depending on its context, which makes it more accurate and versatile. Similar work for studying public opinions on Human Papilloma Virus (HPV) vaccines on social media has been discussed in [10].

Similar to cyberbullying detection, text summarizing has attracted the attention of researchers in the field of NLP[11]. This application is usually performed through two methods, namely, extractive text summarizer and abstractive text summarizer. The paper has focused on retrieving the valuable amount of data using the ELMo embedding in extractive text summarization.

In a recently published paper, the authors have shown the performance of ELMo, where it is applied on a multi-language platform[12]. Similar to other ELMo-based applications, the paper proposes pre-trained embeddings from the popular contextual ELMo model for seven languages, namely, Croatian, Estonian, Finnish, Latvian, Lithuanian, Slovenian, and Swedish. The proposed ELMo model's architecture has three neural network layers, where the first layer is a CNN layer, which operates on a character level. It is followed by two BiLSTM layers, each one consisting of two concatenated LSTMs. Based on the structure of ELMo which is trained on character level and has the ability of handling Out Of Vocabulary words, having a file containing the most common tokens can be useful for training and make the embedding generation more efficient[12]. This paper shows how a proposed method initially designed for a specific language like English may be used for other languages as well.

Toxic context detection has also been studied in [13]. The paper considers embeddings, including BERT and FastText, along with a group of Machine Learning (LR, SVM, DT, RF, XGBoost) and Deep Learning algorithms (CNN, MLP, LSTM). Tokenization, performing basic stemming, and lemmatization techniques are done in the preprocessing phase. In the second phase, various ML algorithms, including Logistic Regression, Support Vector Machine, Decision Trees, Random Forest, and Gradient Boosting, are performed. They merged HASOC'20 and ALONE datasets as one major dataset and performed the evaluations on that. It has been shown that a combination of BERT embedding with CNN gives the best results. It has also shown that CNN understands and efficiently identifies appropriate patterns in the case of small sequences of words and noise in the dataset.

As the importance of word embeddings in the combination of neural-based models, authors in [14] proposed a dense classifier with contextual representations using ELMo to use for classifying crisis-related data on social networks during a disaster. They used real-time Twitter datasets, and analyzed the performance using precision, recall, f1-score, and accuracy. The dense model that they used contains two dense layers, which are a dense layer with Rectifier Linear Unit (ReLU) activation function, and the other one is a dense layer with a softmax function. The proposed combination of the dense classifier with ELMo representations gives better accuracy than the traditional classifier, such as SVM and deep learning classifiers CNN and MLP.

Another use of ELMo in text mining, especially in biomedical text classification, may be referred to [15], where they proposed both deep and shallow network approaches, and their predictions are based on the similarity between extracted features from contextualized representations of the words in their dataset. As the word representations, they considered ELMo and BERT. In addition, they proposed transfer learning by adding a dense layer to the pre-trained ELMo model. Their dataset is from the PubMed repository, which has records including biomedical citations and abstracts in an XML format. As one of their results, the ELMo classifier, in combination with one dense layer, outperforms other methods.

It is normal to have noisy data in NLP as the data is mainly collected from crawling the social media where people write their opinions in different formats and languages. Different type of character and word level methods are used by authors in [16] to simulate setups in which input may be somewhat noisy or different from the data distribution on which NLP systems were trained. They evaluated the performance of well-known deep contextualized word embeddings such as ELMo, BERT, XLNet, and RoBERTa. They used BERT, RoBERTa, and XLNet as both

words embedding generators and classifiers, but the word representations provided by the ELMo were fed into one dense layer. The results suggest that some language models can manage specific types of noise more efficiently than other models. ELMo achieved higher scores than BERT, even XLNet, and RoBERTa on some character-level perturbations.

Deep learning algorithms, coupled with word embeddings in detecting cyberbullying texts, are the topic of much research work[17]. In a matrix of choices, three deep learning algorithms, namely GRU, LSTM, and BiLSTM, in conjunction with word embeddings models, including word2vec, GloVe, Reddit, and ELMO models, are used to examine the effectiveness and accuracy of a possible configuration for cyberbullying detection. Similar to many other research works, data preprocessing steps, including oversampling, is performed on the selected datasets related to social media. A typical dataset in the literature, namely, Formspring. me, has been used for performance evaluation. Form spring. me is basically a social site that provides a platform for users to ask any question to any other users. It consists of 12,772 posts. Based on extensive experimental results, BiLSTM performs best with ELMo in detecting cyberbullying texts. As another performance index, the average time taken for the training of each model has also been measured based on which GRU outperforms compared to other methods.

As another survey on the use of a deep learning model in combination with deeply contextualized word embeddings such as BERT, and ELMo, one may refer to [18]. In this paper, the authors conducted experiments to study both classic and contextualized word embeddings in text classification. As the encoder for the sequence of text, they employed CNN and BiLSTM. They selected four different benchmarking classification datasets with variable average sample lengths, which are 20NewsGroup, The Stanford Sentiment Treebank dataset, the arXiv Academic Paper dataset, and Reuters-21578 (Reuters). In addition, they considered both single-label classification and multi-label classification. This study claims that selecting CNN over BiLSTM for document classification tasks is better than for sentence classification datasets. As the second task in this study, they applied CNN and BiLSTM on both ELMo and BERT. Based on reported results, BERT surpasses ELMo, especially for lengthy datasets. As a comparison with classic embeddings, both achieve improved performance for short datasets, while the improvement is not observed in more extended datasets.

3. METHODOLOGY

In this section, the proposed methodology is described in detail in three stages: pre-processing steps for dataset preparation, then word embedding phase followed by various classification methods.

3.1. Required Pre-processing

One of the most important steps in cyberbullying detection is text pre-processing. The common techniques include stop words and punctuation removal, lemmatization, stemming, and emoticon and URL removal [19]. The stop words are referred to as the most commonly used words in any language, such as articles, prepositions, pronouns, and so on. The next step is to generate the text representation. The embeddings are generated following different feature engineering processes. In this study, some of the stop words are maintained because they can enrich the semantics of the text and make improvements to the results [5]. The two performed pre-processing steps are text conversion to lower case and padding and truncating the sentences to a certain number of words as the neural network models need to have input with the same shape and size.

3.2. ELMo Word Embedding

Having pre-processed text, the input is ready to be fed to the selected embedding model. In this study, we choose the ELMo word embedding proposed by[4]. By using Bi-directional Language Models (BILM), this word embedding provides two passes in its structure, which are forward passed, and backward pass. Unlike the other word embeddings such as Glove and Word2Vec, ELMo uses the complete sentence for generating the representation for a word in the sentence. In this study, for the ML algorithms, the ELMo representations are generated separately using the AllenNLP ELMo library[20]. The ELMo word representations are fed to the ML models as the input. For the DL models, a function was defined for the embedding layer, which used the ELMo embedding function from the TensorFlow hub. The signature parameter of the ELMo function is selected as default because the input type is not tokenized. The output of ELMo word embedding is a tensor with the shape of [batch-size, max-length, 1024]. The max length in this study is selected as 100 words per sentence.

3.3. Classification Methods

In the classification phase, various ML classification techniques are used in this study. We briefly describe each classification method with related models in the next few paragraphs.

For the deep learning classification methods, we used the same models used in [1]. As a general description for all DL models, they all have the same number of layers and are structured with an embedding layer for mapping the input text to the word representations. The last layer for all models is a Dense layer, which provides a single binary label as the result of an input. The sigmoid function is used as the activation function.

The Dense model is comprised of three Dense layers with 1024, 64, and 1 neuron. They can reduce the input size of numerous nodes to a few nodes with weights that can be used to predict the label of the input. This is because they are densely connected layers. The difference between our Dense architecture with the ones in the literature[14], [15], and [16] is in the number of layers and the activation function. As mentioned before, the authors in [14] used two-layer of dense, while in this study, we used three dense layers with a different number of neurons. Moreover, in [14], researchers used softmax as the activation function while we used sigmoid as the activation function. Two other papers used only one dense layer in their studies.

The CNN model has two layers, which perform the filtering operation. With its configuration, it extracts the more important features of the text. The kernel size for the first layer is ten and for the second layer is 5. All the layers in this model have the same number of neurons as mentioned in Dense layers so that better comparison can be performed.

The LSTM model is an updated version of Recurrent Neural Networks (RNN). It uses two LSTM layers to perform the classification. This model uses memory blocks to keep the record of the computations. This can help the model to understand the semantic patterns of historical input data and use them in the currently processed data. As the development of the LSTM model, the BiLSTM model uses the bidirectional LSTM layers, which process the training data in two directions, forward and backward, and pass to LSTM hidden layer, and then the results are combined by a shared output layer.

The remaining ML algorithms investigated in this study are MLP, Decision Tree, and Random Forest. The MLP model is composed of a single layer with 100 nodes. The Decision Tree builds a model where the data is continuously split according to specific parameters. The algorithm starts with a root node and is divided into children nodes according to a given set of rules. The Random

Forest model is composed of multiple Decision Trees. By using the majority votes, it chooses the best output as the final label for the input. The number of decision tree estimators used in this study is 100.

4. COMPARATIVE EVALUATION

In this section, after a brief description of the dataset and the experimental setup, the results of ELMo embedding applied to different groups of ML models are reported. Thereafter, a comparative evaluation of the results obtained in this study and the results provided by [1] is presented.

4.1. Dataset Description

We used the dataset released by the Kaggle competition[6]. This dataset is gathered from Wikipedia comments, which have been manually labelled into six different toxicity classes. The dataset has more than 200K comments presenting the labels for six different toxicity classes, which are toxic, severe toxic, obscene, threat, insult, and identity hate. The original dataset is reported as a strongly unbalanced dataset, and it caused a biased training procedure. The authors in [1] provided balanced datasets for each toxicity class where the datasets have an equal number of toxicity examples and the number of non-toxicity examples. Table 1 shows the number of examples in each dataset.

Table 1. Distribution of six classes

Dataset	Toxic	Severe Toxic	Obscene	Threat	Identity Hate	Insult
Num. Records	**42768**	**3924**	**24280**	**1378**	**22608**	**4234**

4.2. Experiment Setup and Evaluation Metrics

The experiments were run on 5-fold cross-validation, and the selected batch size for each model is 8. The models are trained in 5 epochs, and a binary cross-entropy is selected as the loss function. The optimizer is Adam, with the default learning rate of 0.002 provided by the library. To implement the ML algorithms, we used the Scikit-learn library. All the other parameters are based on the model's performance and previous experiences in the competitor's work. The experiments have been done on Google Colab GPU with High RAM of 26 GB memory.

We report the Precision, Recall, F1-Score, and accuracy of the cyberbullying detection results in this study. The Precision, Recall, and F-score are computed according to Equations 1, 2, and 3, respectively. The parameters used in these equations are True Positive (TP) which shows the number of correct instances guessed by the implemented models, and False Positive (FP), which is the number of false predicted instances by models. Moreover, False Negative (FN), which shows the number of instances erroneously associated with a wrong class is used in Recall equations.

$$Precision(P) = \frac{TP}{TP + FP} \quad (1)$$

$$Recall(R) = \frac{TP}{TP + FN} \quad (2)$$

$$F - measure(f) = 2 \times \frac{P.R}{P+R} \quad (3)$$

4.3. Results and Analysis

In this section, we discuss the results of the experiments that we have performed. The results are divided into three tables which contain the results of the baseline paper [1] and current research results on precision, recall, and F1 score. The authors of [1] compared the effect of TF-IDF word embedding on three ML models. In this study, the effect of ELMo word embedding is evaluated on four deep learning models and three basic machine learning models. We then compare the results against the combination of TF-IDF in the same three basic ML models and the effect of Word2Vec and BERT embeddings on the same four DL models. It is worth mentioning that, since authors in [1] used three different versions of Word2Vec(pre-trained, domain-trained, and Mimicked) and based on their result analysis, the mimicked Word2Vec achieved the best results. Therefore, in this study, we compare our results against the Mimicked Word2Vec.

Table 2. Comparison of precision of ELMo against TF-IDF using three basic ML algorithms on six different datasets

	Feature	Toxic	Sever Toxic	Obscene	Threat	Identity Hate	Insult
Decision Tree	TF-IDF	**0.859**	**0.847**	**0.926**	**0.917**	0.819	**0.887**
	ELMo	0.661	0.751	0.690	0.749	**0.838**	0.701
Random Forest	TF-IDF	**0.860**	0.888	**0.945**	**0.954**	0.847	**0.929**
	ELMo	0.800	**0.890**	0.821	0.865	**0.861**	0.822
MLP	TF-IDF	0.849	**0.913**	0.884	0.914	0.889	0.871
	ELMo	**0.855**	0.901	**0891**	**0.937**	**0.902**	**0.872**

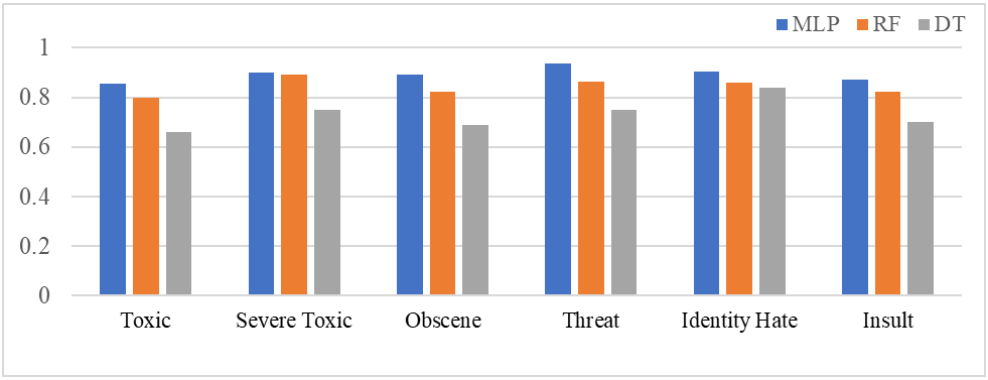

Fig.1. Comparison of ELMo based on precision using ML models

As shown in Table 2, we performed both TF-IDF and ELMo word embedding on MLP, Random Forest, and Decision Tree models. The obtained results show that ELMo outperforms TF-IDF when it is combined with the MLP model on precision. Moreover, we can see from Fig. 1 that ELMo word embedding has the best results on MLP compared to using Random Forest and Decision Tree models.

The reason behind it could be because the structure and functionality of the tree-based models. The tree-based models split features of a dataset and predict the labels in the leaf nodes. Having this fact in mind, the tree-based models can perform better on the datasets that have more features to split the tree based on that attribute. In our case, the only component of the dataset is the text of comments that converts to word representations. This way, the tree-based models do not get to use many attributes, however, still be able to calculate how much the representations are correlated to the labels.

Table 3. Comparison on the precision of ELMo against BERT and mimicked Word2Vec using DL models on six different datasets

	Feature	Toxic	Sever Toxic	Obscene	Threat	Identity Hate	Insult
Dense Model	Mimicked	**0.868**	**0.926**	0.880	**0.933**	**0.881**	0.873
	BERT	0.828	0.912	0.844	0.867	0.874	0.841
	ELMo	0.838	0.845	**0.891**	0.801	0.861	**0.879**
CNN Model	Mimicked	0.836	0.886	0.856	**0.927**	0.860	0.847
	BERT	0.801	0.899	0.819	0.842	0.824	0.831
	ELMo	**0.874**	**0.912**	**0.859**	0.900	**0.888**	**0.860**
LSTM Model	Mimicked	**0.895**	0.941	0.928	0.953	0.887	0.916
	BERT	0.866	0.927	0.889	0.916	0.880	0.874
	ELMo	0.681	**0.962**	**0.943**	**0.970**	**0.943**	**0.961**
BiLSTM Model	Mimicked	**0.910**	0.939	0.929	0.941	0.902	0.920
	BERT	0.875	0.933	0.892	0.913	0.900	0.889
	ELMo	0.680	**0.951**	**0.944**	**0.974**	**0.951**	**0.937**

Among the four DL models that are implemented using three-word embeddings, ELMo embedding outperforms Mimicked Word2Vec and BERT in most categories of CNN, LSTM, and BiLSTM models. Specifically, in the LSTM model, using ELMo word embeddings provided a good improvement in terms of precision with a minimum of 2% and a maximum amount of 5%. The ELMo model does not perform very well on the Toxic dataset among all the models. As mentioned before, the pre-processing steps did not act on these datasets because the punctuations and stop words have effects on the semantics of the sentence. Since the Toxic dataset is the largest, having more irrelevant words are unavoidable. Hence, these results are likely the result of having more stop words such as "the", "is", and so on in the Toxic dataset.

In Tables 4 and 5, we report results on the recall values obtained when different models and embeddings are combined. In general, the results obtained on recall are symmetrically better than precision ones in the combination of ELMo embeddings and all ML models. Based on the definition of precision and recall, higher recall means that the model predicts the most relevant results, and higher precision means that the model returns more relevant results than irrelevant ones. In other words, based on the definition of False Positive and False Negative, which are mentioned in section 4.2, getting a false negative has a much more significant impact than having a false positive in cyberbullying detection because the false negatives in cyberbullying detection mean the bullying comments are predicted as non-bullying ones while the false positives mean the non-bully instances are predicted as bullying contents. The goal of cyberbullying detection is to find and predict the correct bully instances and prevent the occurrence. Therefore, if the model predicts the bully instances as the non-bully ones, then the damage is bigger. Thus, having lower false negatives can help to have better recall due to the nature of this study.

Table 4. Comparison of recall of ELMo against TF-IDF using three basic ML models on six different datasets

	Feature	Toxic	Sever Toxic	Obscene	Threat	Identity Hate	Insult
Decision Tree	TF-IDF	**0.855**	**0.947**	**0.929**	**0.891**	**0.927**	**0.891**
	ELMo	0.657	0.770	0.691	0.791	0.820	0.690
Random Forest	TF-IDF	**0.856**	**0.940**	**0.834**	**0.897**	**0.911**	**0.851**
	ELMo	0.718	0.855	0.760	0.821	0.890	0.752
MLP	TF-IDF	0.857	0.918	**0.895**	0.916	0.897	0.880
	ELMo	**0.859**	**0.927**	0.871	**0.921**	**0.978**	**0.895**

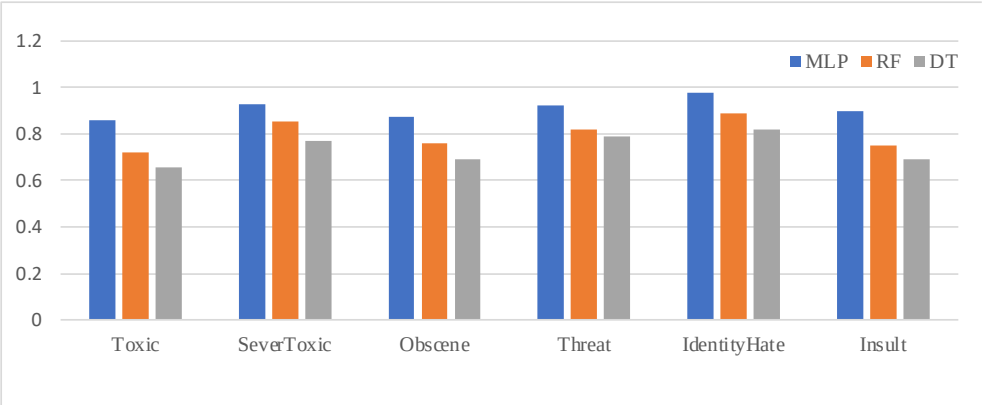

Fig 2. Comparison of ELMo based on Recall using three basic ML models

Similar to what is presented in Table 2, the TF-IDF performs better than ELMo when it is used in Random Forest and Decision Tree models. The combination of ELMo and MLP underperforms slightly compared to using TF-IDF on the Obscene dataset. The comparison between the combination of ELMo and three basic ML models is shown in Fig2. ELMo embedding demonstrated better results only when combined with MLP compared to the integration of ELMo with the other two basic ML models.

Table 5. Comparison of Recall of ELMo against BERT and Mimicked Word2Vec using DL model on six different datasets

	Feature	Toxic	Sever Toxic	Obscene	Threat	Identity Hate	Insult
Dense Model	Mimicked	0.844	0.914	**0.877**	0.932	0.882	0.857
	BERT	0.817	0.917	0.821	0.891	0.865	0.827
	ELMo	**0.905**	**0.929**	0.871	**0.970**	**0.920**	**0.890**
CNN Model	Mimicked	**0.865**	0.919	**0.870**	0.918	0.879	0.849
	BERT	0.812	0.911	0.832	0.872	0.842	0.821
	ELMo	0.857	**0.920**	0.863	**0.899**	**0.884**	**0.880**
LSTM Model	Mimicked	**0.938**	**0.966**	0.938	**0.962**	0.946	0.948
	BERT	0.851	0.932	0.861	0.899	0.895	0.870
	ELMo	0.889	0.949	**0.952**	0.951	**0.982**	**0.952**

BiLSTM Model	Mimicked	**0.921**	0.963	0.945	**0.944**	0.934	0.935
	BERT	0.841	0.941	0.852	0.900	0.857	0.866
	ELMo	0.869	**0.984**	**0.959**	0.953	**0.971**	**0.960**

The results of ELMo on Dense, LSTM, and BiLSTM models are better than Mimicked Word2Vec and BERT. Although authors stated in [1] that the Dense model had the worst results among the other deep learning models, in this study, we found out that the combination of the Dense model with ELMo improves the outcomes against the combination of this model with BERT, mimicked Word2Vec.

Table 6. Comparison of F1-score of ELMo against TF-IDF using three basic ML models on six different datasets

	Feature	Toxic	Sever Toxic	Obscene	Threat	Identity Hate	Insult
Decision Tree	TF-IDF	**0.857**	**0.894**	**0.928**	**0.903**	**0.869**	**0.889**
	ELMo	0.670	0.761	0.700	0.783	0.841	0.719
Random Forest	TF-IDF	**0.858**	**0.913**	**0.913**	**0.924**	0.877	**0.888**
	ELMo	0.761	0.871	0.791	0.846	**0.879**	0.790
MLP	TF-IDF	0.853	0.915	0.889	0.913	**0.893**	0.876
	ELMo	**0.860**	**0.918**	**0.901**	**0.929**	0.881	**0.883**

Table 7. Comparison of ELMo against BERT and Mimicked Word2Vec based on F1-score using DL models on six different datasets

	Feature	Toxic	Sever Toxic	Obscene	Threat	Identity Hate	Insult
Dense Model	Mimicked	0.844	0.914	0.919	0.931	0.880	0.863
	BERT	0.855	0.917	0.913	0.877	0.855	0.834
	ELMo	**0.905**	**0.929**	**0.946**	**0.970**	**0.880**	**0.880**
CNN Model	Mimicked	**0.865**	0.919	0.901	**0.922**	0.869	0.847
	BERT	0.812	0.911	0.904	0.849	0.832	0.826
	ELMo	0.857	**0.920**	**0.918**	0.881	**0.883**	**0.870**
LSTM Model	Mimicked	**0.916**	**0.966**	0.953	0.957	0.914	0.931
	BERT	0.858	0.932	0.929	0.907	0.886	0.872
	ELMo	0.760	0.949	**0.955**	**0.960**	**0.961**	**0.970**
BiLSTM Model	Mimicked	**0.915**	0.963	0.951	0.940	0.916	0.927
	BERT	0.856	0.941	0.937	0.905	0.874	0.877
	ELMo	0.760	**0.980**	**0.970**	**0.960**	**0.960**	**0.950**

Tables 6 and 7 show the results of the F1 score on different word embeddings on different DL models. The combination of MLP and ELMo outperforms all other DL models. From the DL perspective, the BiLSTM model, which has a complex architecture, gets the best results in combination with ELMo. This combination has outdone the others with a minimum improvement of 2% and a maximum improvement of 4%. It is interesting to observe that the combination of ELMo with the Dense model has the best results against BERT and Mimicked word2vec word embeddings in all six datasets. This combination obtains the same result as the combination of the Dense model and Mimicked word2Vec just on the Identity hate dataset, which is still the

highest outcome for this dataset. Again, ELMo does not provide good representations for the Toxic dataset.

From the results, we conclude that the TF-IDF algorithm is a good choice as a word embedding for resources to be parsed with ML models such as Random Forest and Decision Tree. Moreover, the results suggest that ELMo word embeddings could be a good choice for ML algorithms which has a neural network-based, such as MLP, because the structure of ELMo word embeddings is based on a two-layer bidirectional language model which has two passes, forward pass, and backward pass, which solves the problem of polysemy in word representation.

Surprisingly, between BERT and ELMo embeddings, BERT performs worse on this task. The authors in [1] think the reason that caused BERT's undesirable results is that assigning a different embedding to the same word is confusing to the training of the DL models. However, as mentioned above, the strength of ELMo is that it can take the entire input sentence into an equation when calculating the word embeddings. Therefore, the selected word would produce different ELMo vectors in different contexts.

5. CONCLUSION AND FUTURE WORK

In today's age of Information and Communication Technology, the availability of detection systems to prevent the spread of harassment and cyberbullying behaviour promotes a safer and healthier adoption of social media platforms. The core of cyberbullying detection systems is composed of word embedding and classification techniques, for which AI-based solutions are essential. In this paper, we considered ELMo-based methods as word embedding techniques combined with Dense, CNN, LSTM, and the BiLSTM methods as deep learning models and MLP, Random Forest, and Decision Tree as other machine learning classification techniques. The rich datasets from the Kaggle competition were used for performance and comparative evaluations. The practical results show that the combination of ELMo word embedding with most of the deep learning models outperforms other combinations of word embeddings and deep learning models. Moreover, it is interesting to observe that combining ELMo word embedding with MLP, which is a neural network-based model, produces better results than other machine learning algorithms. For future work and as a necessary step toward a real-life application of cyberbullying detection, we will investigate, in the immediate future, the use of an online scheme for ELMo word embedding and classification.

ACKNOWLEDGEMENT

We gratefully acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN-2020-06482.

REFERENCES

[1] D. Dessì, D. R. Recupero, and H. Sack, (2021) "An Assessment of Deep Learning Models and Word Embeddings for Toxicity Detection within Online Textual Comments," *Electronics*, vol. 10, no. 7, p. 779, doi: 10.3390/electronics10070779.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, (2013) "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems*, vol. 26. Accessed: May 31, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, (2019) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[4] M. E. Peters *et al.*, (2018) "Deep Contextualized Word Representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, pp. 2227–2237. doi: 10.18653/v1/N18-1202.

[5] H. H. Saeed, K. Shahzad, and F. Kamiran, (2018) "Overlapping Toxic Sentiment Classification Using Deep Neural Architectures," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Singapore, Singapore, pp. 1361–1366. doi: 10.1109/ICDMW.2018.00193.

[6] "Toxic Comment Classification Challenge." https://kaggle.com/c/jigsaw-toxic-comment-classification-challenge (accessed Sep. 22, 2021).

[7] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C. J. Kuo, (2022) "Evaluating word embedding models: methods and experimental results," *APSIPA Trans. Signal Inf. Process.*, vol. 8, no. 1, 2019, doi: 10.1017/ATSIP.2019.12.

[8] D. S. Asudani, N. K. Nagwani, and P. Singh, (2021) "Exploring the effectiveness of word embedding based deep learning model for improving email classification," *Data Technol. Appl.*, doi: 10.1108/DTA-07-2021-0191.

[9] K. S. Kalyan and S. Sangeetha, (2020) "SECNLP: A survey of embeddings in clinical natural language processing," *J. Biomed. Inform.*, vol. 101, p. 103323, doi: 10.1016/j.jbi.2019.103323.

[10] L. Zhang, H. Fan, C. Peng, G. Rao, and Q. Cong, (2020) "Sentiment Analysis Methods for HPV Vaccines Related Tweets Based on Transfer Learning," *Healthcare*, vol. 8, no. 3, p. 307, doi: 10.3390/healthcare8030307.

[11] H. Gupta and M. Patel, (2020) "Study of Extractive Text Summarizer Using The Elmo Embedding," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, pp. 829–834. doi: 10.1109/I-SMAC49090.2020.9243610.

[12] M. Ulčar and M. Robnik-Šikonja, (2020) "High Quality ELMo Embeddings for Seven Less-Resourced Languages," in *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France, pp. 4731–4738. Accessed: May 31, 2022. [Online]. Available: https://aclanthology.org/2020.lrec-1.582

[13] P. Malik, A. Aggrawal, and D. K. Vishwakarma,(2021) "Toxic Speech Detection using Traditional Machine Learning Models and BERT and fastText Embedding with Deep Neural Networks," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, pp. 1254–1259. doi: 10.1109/ICCMC51019.2021.9418395.

[14] S. Madichetty and S. M, (2020) "Improved Classification of Crisis-Related Data on Twitter using Contextual Representations," *Procedia Comput. Sci.*, vol. 167, pp. 962–968, doi: 10.1016/j.procs.2020.03.395.

[15] D. A. Koutsomitropoulos and A. D. Andriopoulos, (2022) "Thesaurus-based word embeddings for automated biomedical literature classification," *Neural Comput. Appl.*, vol. 34, no. 2, pp. 937–950, doi: 10.1007/s00521-021-06053-z.

[16] M. Moradi and M. Samwald, (2021) "Evaluating the Robustness of Neural Language Models to Input Perturbations," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, pp. 1558–1570. doi: 10.18653/v1/2021.emnlp-main.117.

[17] M. Al-Hashedi, L.-K. Soon, and H.-N. Goh, (2019) "Cyberbullying Detection Using Deep Learning and Word Embeddings: An Empirical Study," in *Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems*, Bangkok Thailand, pp. 17–21. doi: 10.1145/3372422.3373592.

[18] C. Wang, P. Nulty, and D. Lillis, (2020) "A Comparative Study on Word Embeddings in Deep Learning for Text Classification," in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, New York, NY, USA, pp. 37–46. doi: 10.1145/3443279.3443304.

[19] D. Dessì, G. Fenu, M. Marras, and D. Reforgiato Recupero, (2019) "Bridging learning analytics and Cognitive Computing for Big Data classification in micro-learning video collections," *Comput. Hum. Behav.*, vol. 92, pp. 468–477, doi: 10.1016/j.chb.2018.03.004.

[20] "AllenNLP - ELMo — Allen Institute for AI." https://allenai.org/allennlp/software/elmo (accessed May 29, 2022).

AUTHORS

Tina Yazdizadeh

Tina is a Master of Information Technology with a specialization in Data Science student at Carleton University, Ottawa, Canada. Her current research is focused on the intersection of the very demanding fields, namely, "Text Mining" and "Cybersecurity". Before joining the Department of Information Technology at Carleton University, she had received her B.Sc. in Computer Engineering (Software) from the University of Tehran. Her B.Sc thesis was on Map Matching Using GPS Data, a research work which was supported by TAPSI Co. as a growing E-Taxi company very similar to Uber.

Wei Shi

Dr. Wei Shi is a Professor in the School of Information Technology, cross-appointed to the Department of Systems and Computer Engineering in the Faculty of Engineering & Design at Carleton University. She specializes in algorithm design and analysis in distributed environments such as Data Centers, Clouds, Mobile Agents, Actuator systems, and Wireless Sensor Networks. She has also been conducting research in data privacy and Big Data analytics. She holds a Bachelor of Computer Engineering from Harbin Institute of Technology in China and received her master's and Ph.D. in Computer Science from Carleton University in Ottawa, Canada. Dr. Shi is also a Professional Engineer licensed in Ontario, Canada.

.