A K-Competitive Autoencoder for Aggression Detection in Social Media text

Promita Maitra
Mumbai, India
promita.maitra@gmail.com

Ritesh Sarkhel
Ohio State University
Columbus, Ohio
sarkhel.5@osu.edu

Abstract

We present an approach to detect aggression from social media text in this work. A winner-takes-all autoencoder, called Emoti-KATE is proposed for this purpose. Using a log-normalized, weighted word-count vector at input dimensions, the autoencoder simulates a competition between neurons in the hidden layer to minimize the reconstruction loss between the input and final output layers. We have evaluated the performance of our system on the datasets provided by the organizers of TRAC workshop, 2018. Using the encoding generated by Emoti-KATE, a 3-way classification is performed for every social media text in the dataset. Each data point is classified as 'Overtly Aggressive', 'Covertly Aggressive' or 'Non-aggressive'. Results show that our proposed method is able to achieve promising results on some of these datasets. In this paper, we have described the effects of introducing an winner-takes-all autoencoder for the task of aggression detection, reported its performance on four different datasets, analyzed some of its limitations and how to improve its performance in future works.

1 Introduction

With the rapid growth of unregulated social media platforms, a major problem coming to surface is the aggressive nature of text used by people while interacting in these mediums. Manual monitoring or filtering of this user generated data is a challenging task due to its sheer scale. Therefore, automatic detection of aggressive text is the logical first step to combat the issue. There has been an increased interest among contemporary researchers to propose an acceptable solution of this problem in recent years. However, proposing an automated solution for this problem has some inherent obstacles. One of the most challenging obstacles among this to annotate these texts with an appropriate sentiment score. Social media texts are almost always short in length, and exhibit ambiguous grammatical syntax including abbreviated forms, typo errors, repeated alphabets to emphasize intentions as well as coinage of new words. Additionally, detecting aggression on social media posts or comments is especially challenging due to its lack of context. Almost all of these posts are provide very little to no context. Detecting common 'hate-words' using a bag-of-words analysis does not work in these cases as conventionally non-aggressive words can be deemed aggressive when used sarcastically.

In this paper we have proposed Emoti-KATE, a winner-takes-all autoencoder for representing social media text. Performance of our autoencoder is evaluated on a downstream task of classifying the text based on its aggression level. We have broadly categorized a social media post or comment into one of three categories: *Overtly Aggressive*, *Covertly Aggressive* and *Non-Aggressive*. In recent times, researchers (Socher et al., 2011b; Hermann and Blunsom, 2013; Chen and Zaki, 2017) have established that autoencoders can be effectively used to learn representations of document text for various applications. However, they have some inherent limitations. As Chen et al. (Chen and Zaki, 2017) has pointed out, in a traditional autoencoder, contributions of most of the hidden neurons in reconstructing the input vector are often redundant. They have shown that introducing competition among these hidden neurons can help get rid of these redundancies. The output vectors generated by such winner-takes-all approach

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/

Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, pages 80–89
Santa Fe, USA, August 25, 2018.

outperform most of the popular, contemporary document representation techniques as well. Inspired by their success, we have taken a similar approach in this work. Emoti-KATE introduces a K-competition layer between the input and output layers to generate the vector representation of each social media text. We have evaluated the performance of our approach on a dataset of 15,000 aggression-annotated Facebook posts and comments each in Hindi (in both Roman and Devanagari script) and English. After basic preprocessing on the raw data[1], the winner-takes-all autoencoder is deployed with a log-normalized vector at its input dimensions. Results show that while our system's performance is promising in case of English texts (weighted F1 score of 0.5694), the classifier has much scope of improvement for Hindi texts (weighted F1 score of 0.4189). The main contribution of this work is an exhaustive investigation into the performance of K-competitive autoencoders (Chen and Zaki, 2017) in identifying aggression level in short, sparsely contexualized social media posts and comments. Our experimental result suggests that introducing a competitive hidden layer in a autoencoder framework improves the performance of aggression detection in sparse social media texts. A complete pseudocode of our system is presented in Algorithm 1.

The rest of the paper is organized as follows: Section 2 gives a brief overview of the work already done in this field. Section 3 describes the details of the dataset, preprocessing steps and the system we have used to address the problem. Section 4 presents the analysis of results and finally, the conclusion and future scopes are discussed in Section 5.

2 Related Work

Automatically detecting cyber-bullying and use of hate speech from textual analysis has gained momentum with the rise of increasing amount of user-generated content on web. Most of the research works have considered this to be a binary classification problem, i.e. aggressive or non-aggressive. The set of features that has been widely used by contemporary researchers Schmidt and Wiegand (2017) includes various word-level and character-level features including n-grams, usage of punctuations, capitalization, and token-length. Some recent approaches have also proposed sentiment polarity detection, using lexical resources available in web along with bag-of-words, and linguistic features such as POS tagging, dependency parsing, knowledge-based feature extraction using ontology information etc. for this purpose. To solve the problem of context sparsity, researchers have proposed some word clustering techniques as well as meta-data analysis including user-profile analysis, and user-activity history analysis etc. However, most of these existing works have some strong assumptions inherent to their systems. For example, in Dinakar et al. (2011) the authors assumed that an aggressive post in social media can primarily be about one of the few topics, which include physical appearance, sexuality, race and culture, and intelligence. They trained a multi-class classifier for identifying texts on these topics and individual binary classifiers to subsequently classify whether the text covered one of the topics mentioned above. Features such as TF-IDF, presence of swear words, frequent bigrams, and topic-specific n-grams were used to train the classifiers. Dadvar et al. (2013) followed a similar approach and introduced user-profile specific meta-data (separate classifiers based on user gender), which helped them to improve the precision of their system. Nahar et al. (2012) extracted semantic features using Latent Dirichlet Allocation (LDA) along with lexicon features, TF-IDF values and second-person pronouns to train a Support Vector Machine for classification purposes. Sentiment analysis technique was leveraged to detect cyberbullying in a Twitter dataset by Xu et al. (2012). They used Latent Dirichlet Allocation to identify the most frequent topics in tweets that exhibited signs of bullying. Semi-supervised approaches with bootstrapping have also been proposed in recent years (Schmidt and Wiegand, 2017). While there have been many significant research contributions in this area, most of them considered the problem of aggression detection as a binary classification task, i.e., each text was to be classified either as aggressive or non-aggressive. Malmasi and Zampieri (2018) is one of the early works that has taken a different approach. In their work, every social media text is classified into three categories, hate speech, aggressive text and neutral. They argued that as general profanity is an unavoidable part of social media posts due to their unregulated nature, our efforts

[1] we observed that 'hashtags' carry significant information in social media text and the segmentation improved our prediction score on validation set by 3% to 5%

should focus on distinguishing profanity from hate speech that targets an individual or a group. They have used a single as well as ensemble classifiers with stacked generalization for this purpose. Their feature set includes n-grams, skip-grams and clustering-based word representations.

With the evolution of deep neural network (DNN) based models for natural language analysis, there has been some works using leveraged DNN models for this purpose as well. For example, Mehdad and Tetreault (2016) have proposed a Recurrent Neural Network based Language Model approach with character n-gram feature for this task to overcome the challenge of scarcity of large-scale dataset. Zhao et al. (Zhao and Mao, 2017) have recently proposed a semantically enhanced marginalised denoising autoencoder for detecting cyberbullying on social media text. In this work, they have extended a stacked denoising autoencoder with semantic dropout noise and sparsity constraints. Contrary to these methods, we have used a shallow autoencoder in Emoti-KATE. Following the work of Chen and Zaki (2017), we have introduced a K-competition layer in our framework to tackle the problem of data sparsity and little contextual information in these texts. This layer reduces the redundancy of reconstructed input patterns in the respective contributions of hidden neurons to the final output layer. The main contribution of our work is an exhaustive investigation of winner-takes-all autoencoder framework in detecting aggression level in sparse social media texts. Our model has been evaluated on datasets collected from Facebook and Twitter, in both Hindi and English languages.

Algorithm 1: K-competitive Autoencoder

Input: Training set (D_{train}), Test set (D_{test})
Output: Feature vectors (V_{test}) of D_{test}

Initialize $O_{test} \leftarrow \Phi$ ▷ O_{test} ←Autoencoder output vectors for D_{test}

for *each document* $d \epsilon \{D_{train} \cup D_{test}\}$ **do**
| Compute input vector v_d
end

$W^{'} \leftarrow$ `training`(V_{train})
$O_{test} \leftarrow \tanh(W^{'} V_{test} + b)$ ▷ $V_{test} \leftarrow$ Input vectors of D_{test}
return O_{test}

procedure `training`: ▷ V_{train} ←Input vectors for D_{train}
Forward propagate $z = \tanh(W V_{train} + b)$
Apply K-competition on activations $z^{'}$ = `K-compete`(z)
Compute error, back-propagate and iterate until convergence
return W;

3 Methodology and Dataset

A K-competitive autoencoder for aggression detection in social media text is proposed in this work. A detailed description of the dataset, some of its interesting characteristics and our approach towards this task will be presented in this section.

3.1 Description of the dataset

The aggression-annotated dataset used in this shared task(Kumar et al., 2018a) has been collected and prepaed by Kumar et al. (2018b). Collecting approximately 18k tweets and 21k facebook comments from social media users in India, they used Crowdflower, a crowd-sourced platform to annotate the entire dataset.[2] This dataset is code-mixed i.e., it contains text in English and Hindi (written in both Roman and Devanagari script). A total of 3 top-level and 10 level-2 tags were assigned to the entire dataset. In this shared task, we have only considered the top-level tags, which are as follows: 'Overtly Aggressive', 'Covertly Aggressive', and 'Non-aggressive'. The organizers of TRAC 2018 have divided

[2]Interested readers can find more details about the data collection methods used to compile this dataset in Kumar et al. (2018b).

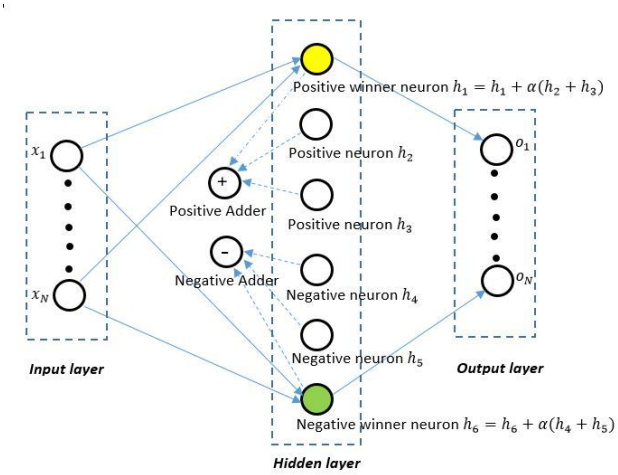

Figure 1: **An illustration of competition among the hidden neurons in Emoti-KATE; The hidden layer shows only one of each positive and negative neurons are selected as the winners out of the six neurons competing to contribute to the output layer; All of the neurons in input, hidden and output layers are fully connected but not shown in this figure for ease of interpretation**

this dataset into a number of smaller datasets based on their origin (social media platform from where it was collected) and language. This resulted in 4 different datasets for this task, which are as follows, *English-Facebook* dataset, *English-Social Media* dataset, *Hindi-Facebook* dataset, and *Hindi-Social Media* dataset. We have evaluated and reported the performance of *Emoti-KATE* in Section 4, for each of these datasets.

3.2 Preprocessing steps

Depending on the dataset, each document undergoes a number of preprocessing steps. The stopwords were removed first. Then, the documents were stemmed using Porter Stemmer. We have used the NLTK library for both of these steps. Next, each document representing a post in Twitter was further processed as the 'hashtags' used along the normal text were further segmented into meaningful words. For example, "*#savethegirlchild*" is segmented into four distinct words "*save*", "*the*", "*girl*", and "*child*" at the end of this step. If there are multiple such meaningful segmentations possible, all of these combinations are generated. The segmented words generated by this process is then appended to the rest of the text in the document, in the order they appeared in the original 'hashtag' itself. We have used data-driven exhaustive search within the Brown corpus (Marcus et al., 1993) for this purpose. Split points are decided by iteratively searching for every possible combination of meaningful words within the corpus. The split points that produced the highest percentage of meaningful words are considered to be a possible segmentation of the 'hashtag'. We observe that although this strategy of handling 'hashtags' provides the highest recall, it is very slow. Additionally, for social media platforms such as Twitter where the maximum number of characters is limited and a significant number of posts use code-mixed data, this strategy do not work very well.

3.3 Description of Emoti-KATE

Autoencoders (Vincent et al., 2008) are neural networks tasked to reconstruct an input vector by minimizing the loss between the input and the final output layer. In recent years, it has been successfully used to extract features encoding text data (Socher et al., 2011a; Li et al., 2015; Yang et al., 2017). We have used a K-competitive autoencoder, called *Emoti-KATE* in this work. Contrary to a traditional shallow autoencoder, a K-competitive autoencoder (Chen and Zaki, 2017), introduces a competition layer among the hidden neurons. It has been shown in Chen and Zaki (2017), when encoding a sparse text document, this competition among hidden neurons helps get rid of the redundancy contributed by some of the hid-

den neurons in the final encoding. In the K-competitive layer, neurons compete among themselves for the right to respond to the input vector, therefore focusing more on unique and important patterns in the input data. Complete pseudo-code of the K-competitive encoder used in this work is presented in Algorithm 1. Emoti-KATE is a shallow autoencoder, with a single competitive hidden layer. Let, $x \epsilon \text{Re}^d$ denotes the d-dimensional input vector of the autoencoder. The objective of this autoencoder is to reconstruct x at the output layer. Let, $h_1, h_2, ...h_n$ denotes the neurons at the hidden layer. The weights between the input-to-hidden layers and hidden-to-output layers are tied together. Therefore, if $W \epsilon \text{Re}^{d \times n}$ represents the input-to-hidden layer weight matrix, $W^T \epsilon \text{Re}^{n \times d}$ will denote the weight matrix between hidden-to-output layer. We have used tanh(x) $= \frac{e^{2x}-1}{e^{2x}+1}$ as the activation function between the input-to-hidden layer and sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ as the activation function between hidden-to-output layer in our implementation.

Input vector	**Description**
X_1	Word-counts weighted by sentiment-score
X_2	X_1 augmented with character-case information
X_3	X_2 with 'hashtags'-segmented into unique words

Table 1: **A brief overview of the input vectors used in our implementation**

In this work, we have experimented with three different input vectors. A brief overview of how these input vectors are computed is presented in Table 1. As mentioned, the first representation (X_1) denotes a weighted word-count vector of the document. Each word count is weighted by the sentiment score for that word. We have used the sentiment analysis module embedded in *nltk*[3] for this purpose. The second input vector representation (X_2) appends X_1 with character-case statistics of the document. More specifically, X_1 is appended with a vector $C = c_1c_2c_3$ of length 3, where c_1, c_2 and c_3 denote the number of lower, upper, and camel case words in the document. Finally, the last input vector (X_3) follows the same steps as X_2, except one preprocessing step. The main difference between X_2 and X_3 is based on how they process 'hashtags' in a document. In case of both X_1 and X_2 they are treated as distinct, individual words, whereas X_3 takes a different approach. Each 'hashtag' in the document is segmented into distinct meaningful words. The steps followed for this purpose has been described in Section 3.2. Following the implementation by Chen et al. (Chen and Zaki, 2017), each input vector was log-normalized before it is fed into the autoencoder. The training and encoding steps of Emoti-KATE have been shown in Algorithm 1. We have used cross-entropy loss to compute the backpropagation error in our implementation.

The main difference of a K-competitive autoencoder from a traditional shallow autoencoder is a K-competition hidden layer. As mentioned before, we have used a single K-competitive layer in this work. In a K-competitive layer with n neurons, say $h_1, h_2, ...h_n$, gradients flowing through only the top-K neurons obtain the right to contribute to the output layer. In Emoti-KATE, this selection of K neurons is performed based on the activation values of the hidden neurons. We select the top $\lceil \frac{K}{2} \rceil$ positive neurons and the lowest $\lfloor \frac{K}{2} \rfloor$ negative neurons during each feedforward step. The rest of the $n - K$ loser neurons are zeroed out. To compensate for the loss of these neurons as well as to amplify the competition among the neurons, the winner neurons are then rewarded. This means reallocating the loss of activations of the loser neurons to the winners therefore amplifying them in the process. We have illustrated the workflow of the K-competition layer in Fig 1. Activations from the hidden neurons with positive activations i.e., h_2 and h_3 are suppressed and their activations are reallocated to h_1, essentially amplifying its contribution to the output neurons. Therefore, the only positive neuron that contributes to the output layer in Fig. 1 is h_1, as $h_2, h_3 \rightarrow 0$. α is called the amplification factor. It decides how much of the loser activations flow through the winners. For example, if $\alpha > \frac{2}{K}$, gradients flowing through loser neurons are amplified, whereas if $\alpha = 0$, they are completely suppressed and Emoti-KATE effectively (Chen and Zaki, 2017) turns into a K-sparse autoencoder (Makhzani and Frey, 2013). The same process is repeated for the negative neurons too. Hidden neurons h_4 and h_5 are zeroed out and their activations are reallocated to

[3]https://www.nltk.org/api/nltk.sentiment.html

Dataset	Input Vector	Description	F1 (weighted)
English-FB	X_1	wordcount (WC) + sentiment score (SC)	0.5431
	X_2	WC + SC + capitalization feature (CF)	0.5285
	X_3	WC + SC + CF + hashtag analyzer (HA)	0.5694
		Random Baseline	0.3535
English-Social Media	X_1	wordcount (WC) + sentiment score (SC)	0.3379
	X_2	WC + SC + capitalization feature (CF)	0.2884
	X_3	WC + SC + CF + hashtag analyzer (HA)	0.3191
		Random Baseline	0.3477
Hindi-FB	X_1	wordcount (WC) + sentiment score (SC)	0.3969
	X_2	WC + SC + capitalization feature (CF)	0.3911
	X_3	WC + SC + CF + hashtag analyzer (HA)	0.4189
		Random Baseline	0.3571
Hindi-Social Media	X_1	wordcount (WC) + sentiment score (SC)	0.2668
	X_2	WC + SC + capitalization feature (CF)	0.3143
	X_3	WC + SC + CF + hashtag analyzer (HA)	0.3142
		Random Baseline	0.3206

Table 2: **Results of Emoti-KATE on experimental datasets**

the negative winner neuron h_6 with α amplification. As a result, h_6 is the only negative neuron that contributes to the output layer.

4 Results

In this section, we will evaluate Emoti-KATE in learning meaningful representations of social media text in detecting aggression in four different social media datasets. All of our experiments were performed on a 2.80 GHz Intel i7 with 32GB RAM. We have used Keras[4], a high level neural network library for implementing the autoencoder. In our implementation for Emoti-KATE, we have closely followed the design choices by Chen et al.[5] (Chen and Zaki, 2017).

The system was evaluated on the basis of weighted macro-averaged F-scores. The individual F-score achieved for class i.e., Overtly Aggressive (OAG), Covertly Aggressive (CAG) and Non-Aggressive (NAG) was weighted by the proportion of the concerned class in the test set and the final F-score is the average of these individual F-scores of each class. Our system achieved best result for English Facebook texts and the lowest score was in Hindi Twitter data. A detailed description of the results for each dataset has been provided below. The best performance achieved by Emoti-KATE for each of these datasets has been highlighted in the respective tables. The best performance achieved by randomly assigning class labels (random baseline) for each of the datasets have also been reported.

4.1 Results on the English-Facebook dataset

Our system performed best in English Facebook texts with a weighted F1 score of 0.5694 for the input vector X_3, i.e., the input vector that considered Sentiment Score weighted Word Count vector augmented with capitalization feature and 'hashtag' analyzer. The primary reason of our system performing better on this particular dataset is the length of the texts. Typically, texts present in this dataset were longer offering more context, resulting in a less sparse vector which ultimately improved the performance of the classifier. Performance of Emoti-KATE on this dataset, for three different variations of input vectors is presented in Table 2. We have also reported a class wise performance analysis for the best performance (X_3) of our system in Table 3. A heatmap representing the confusion matrix of this run is presented in Fig. 2.

[4]https://github.com/keras-team/keras

[5]https://github.com/hugochan/KATE

Class	Precision	Recall
OAG	39.216	39.216
CAG	33.803	22.967
NAG	72.063	80.639

Table 3: **Class-wise distribution of Precision and Recall for the best result observed (X_3)**

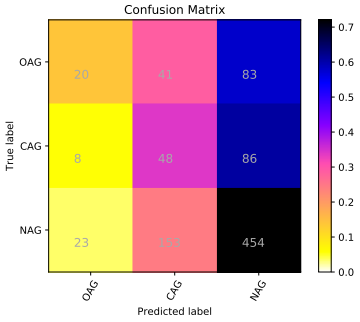

Figure 2: **Heatmap of the confusion matrix for our best result observed (X_3)**

4.2 Results on the English-Social Media dataset

Even though our approach achieved promising score in one of the English datasets, for the English social media (Twitter) dataset, it has significant scope of improvement. The best run for this dataset could only achieve a weighted F1 score of 0.3379. Two main reasons for this are as follows: (1) social media texts, specially tweets consists of a lot of abbreviated forms, repeating characters in a word, newly coined terms etc. which was not efficiently normalized, therefore not present during the training of the classifier, and (2) length of the texts in this dataset was also very short which eventually resulted in sparse vectors. Performance of our system for three different variations of input vectors has been described in Table 2. We have also reported a class wise performance analysis for the best performance (X_1) of our system in Table 4. The heatmap representing a confusion matrix for this run is presented in Fig. 3.

For both the datasets in English, one interesting observation that came up from the confusion matrix is that the system is biased towards the class NAG and this resulted in achieving higher recall in that particular class (highest 80.639%).

4.3 Results on the Hindi-Facebook dataset

Similar to English, our system performs better in classifying Hindi Facebook texts than the social media dataset (Twitter) by achieving a weighted F1 score of 0.4189, which is higher than the random baseline. However, this dataset failed to score at par with the English dataset mainly due to the lack of good quality sentiment lexicon resource in Hindi.

Performance of Emoti-KATE on this dataset, for three different variations of input vectors is presented in Table 2. We have also reported a class wise performance analysis for the best performance (X_3) of our system in Table 5. A heatmap representing the confusion matrix of this run is presented in Fig. 4.

Class	Precision	Recall
OAG	43.137	6.094
CAG	35.319	20.097
NAG	43.46	87.371

Table 4: **Class-wise distribution of Precision and Recall for the best result observed (X_1)**

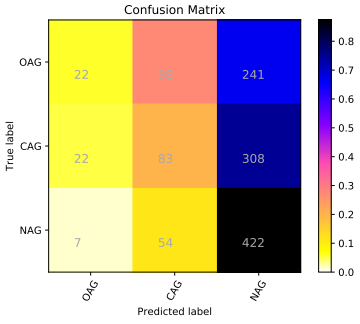

Figure 3: **Heatmap of the confusion matrix for our best result observed (X_1)**

Class	Precision	Recall
OAG	46.988	32.32
CAG	46.379	72.881
NAG	31.944	11.795

Table 5: **Class-wise distribution of Precision and Recall for the best result observed (X_3)**

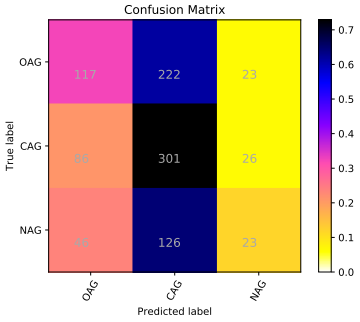

Figure 4: **Heatmap of the confusion matrix for our best result observed (X_3)**

4.4 Results on the Hindi-Social Media dataset

Similar to English-Social Media dataset, there is a lot of scope to improve the performance of our system on the Hindi-Social Media dataset. We observed that unlike the English dataset, the inclusion of the 'hashtag' analyzer module did not improve the performance of this particular dataset. This is because (1) 'hashtags' only consisted about 10 to 30 percent of the entire text on average which was not enough to offset the performance for the entire text content, and (2) the 'hashtag' segmentation module did not leverage a code mixed corpus to generate all possible unique segmentations for this dataset. Training Emoti-KATE on sufficiently large code mixed corpus for 'hashtag' analysis as well as feature vector generation mark some of the most significant future scopes of research. Performance of our system on this dataset, for three different variations of input vectors is presented in Table 2. Similar to the rest of the datasets, We have reported a class wise performance analysis for the best performance (X_2) of our system in Table 6 for this dataset also. A heatmap representing the confusion matrix of this run is also presented in Fig. 5. We observe that our system is biased towards the class CAG in Hindi datasets. Interestingly, highest recall values were obtained for the class CAG for both of the Hindi datasets in our experimental setup.

We have compared our performance against two traditional approaches, a word2vec-based input vector, and an LDA-based input vector. We noticed an average improvement of 13.62% and 16.25% respectively for Word2vec and LDA for all of our training datasets. To investigate the contribution of the competitive hidden layer for the aggression detection task, we have also compared our performance against a traditional variational autoencoder. We observed an average improvement of 19.47% across all datasets. This boost in performance stems from the removal of redundant, confounding contributions made by some of the hidden nodes which were removed by the competitive hidden layer. The differences between output vectors reconstructed by a traditional autoencoder and Emoti-KATE has been illustrated

Class	Precision	Recall
OAG	37.586	23.747
CAG	33.424	64.829
NAG	32.727	15.254

Table 6: **Class-wise distribution of Precision and Recall for the best result observed (X_2)**

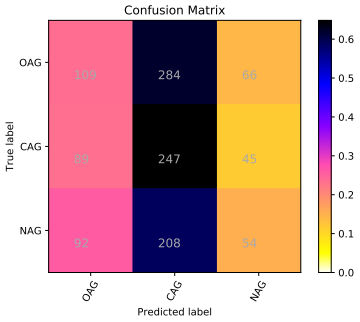

Figure 5: **Heatmap of the confusion matrix for our best result observed (X_2)**

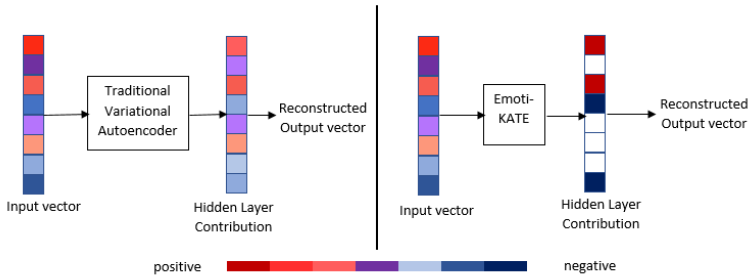

Figure 6: **An illustration of the differences between reconstructed output vectors by a traditional Variational Autoencoder and Emoti-KATE**

in Fig. 6 for better understanding. Additionally, downstream effects of this competitive layer is more pronounced in our results due to the short length nature and contextual sparsity, often observed in social media texts.

It was observed that the system failed to identify a few aggressive sentences mainly due to the absence of a few words in our training vocabulary which contributed to the aggression score. To cite one example, our system predicted the sentence *"Saala darpok. When he comes to south he comes with full security. He makes early morning visit and runs away from backdoor :D"* to be 'NAG', while the correct classification is 'CAG'. For English texts, we trained our system with standard dictionary words and common english slangs, but code-mixed nature of dataset resulted in a decline in performance.

On the other hand, one can notice from the dataset specific precision-recall tables, that we achieved promising result in case of texts which are covertly aggressive, such as *"I have seen rajamouli and his brahman uncle cleaning toilet... and a dalit is head priest in tirupati...lol mahismati. .. this flim also brahmanwadi agenda...bahubali got poonool..99.99% PRAJA are shudra. .."* or *"Or hum ne 1971 se 2017 tak 90k se zeyada tumaray soldiers mar diye"*. Even though these sentences do not explicitly contain any word which can generally be tagged as aggressive, our system was able to detect the sense unlike the traditional autoencoder which tagged these as non-aggressive. While the traditional variational autoencoder failed to capture this category of texts successfully, the introduction of competitive hidden layer helped us achieve a better precision for 'CAG' tagged texts across all domains owing to the removal of loser neurons' contribution as shown in 6.

5 Conclusion

We have proposed an approach for aggression detection from social media text in this work. Using the feature vectors generated by Emoti-KATE, a shallow winner-takes-all autoencoder, a 3-way classification was performed on the experimental datasets, classifying each data point as 'Overtly Aggressive' (OAG), or 'Covertly Aggressive' (CAG) or 'Non-aggressive' (NAG). One of the main challenges of this task was identifying Covertly Aggressive texts. We observed that distinguishing CAG from the rest of the categories (especially NAG) is a difficult task. Handling code switch and transliteration in the Hindi datasets as well as presence of out-of-domain test data in the Social Media datasets made this task more challenging. Results show that our method is a little biased towards the class NAG for English datasets, and the class CAG for the Hindi ones. We also observed that NAG and CAG classes formed the largest set of accurately[6] classified categories in English and Hindi datasets respectively. Although we have achieved promising results for the English datasets, the performance on Hindi datasets can be significantly improved in future works. Using appropriate Hindi corpus for sentiment scoring, training our model on English-Hindi code-mixed data as well as readily available out-of-domain datasets may help achieve better performance than the present system. Additionally, instead of a shallow autoencoder like Emoti-KATE, investigating the performance of competitive stacked denoising autoencoders is also an exciting future endeavor.

[6]based on both precision and recall

References

Yu Chen and Mohammed J Zaki. 2017. Kate: K-competitive autoencoder for text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 85–94. ACM.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *Advances in Information Retrieval*, pages 693–696. Springer.

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

Karl Moritz Hermann and Phil Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 894–904.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018a. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)*, Santa Fe, USA.

Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018b. Aggression-annotated Corpus of Hindi-English Code-mixed Data. In *Proceedings of the 11th Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Alireza Makhzani and Brendan Frey. 2013. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words? In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303, Los Angeles, CA, USA.

Vinita Nahar, Li Xue, and Pang ChaoyiXu. 2012. An effective approach for cyberbullying detection. In *Communications in Information Science and Management Engineering*.

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems*, pages 801–809.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011b. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. *arXiv preprint arXiv:1702.08139*.

Rui Zhao and Kezhi Mao. 2017. Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder. *IEEE Transactions on Affective Computing*, 8(3):328–339.