LSTMs with Attention for Aggression Detection

Nishant Nikhil
IIT Kharagpur
Kharagpur India
nishantnikhil@iitkgp.ac.in

Ramit Pahwa
IIT Kharagpur
Kharagpur India
ramitpahwa123@iitkgp.ac.in

Mehul Kumar Nirala
IIT Kharagpur
Kharagpur India
mehulkumarnirala@iitkgp.ac.in

Rohan Khilnani
IIT Kharagpur
Kharagpur India
rkhilnani9@iitkgp.ac.in

Abstract

In this paper, we describe the system submitted for the shared task on Aggression Identification in Facebook posts and comments by the team Nishnik. Previous works demonstrate that LSTMs have achieved remarkable performance in natural language processing tasks. We deploy an LSTM model with an attention unit over it. Our system ranks 6th and 4th in the Hindi subtask for Facebook comments and subtask for generalized social media data respectively. And it ranks 17th and 10th in the corresponding English subtasks.

1 Introduction

In recent years, there has been a rapid growth in social media usage. Interactions over the web and social media have seen an exponential increase. While usage of social media helps users stay connected; incidents of aggression, trolling, cyberbullying, flaming, and hate speech are more prevalent now than ever.

Recent works on aggression classification include the use of logistic regression classifier (Davidson et al., 2017). They create a bunch of hand-crafted features like binary and count indicators for hashtags, lexicon based sentiment scores for each tweet, unigram, bigram, and trigram features. They use two logistic regression models, the first one to reduce dimensionality of the features and the second one to make classification. Kwok and Wang (2013) train a binary classifier to label tweets into 'racist' and 'non-racist'. They deploy Naive Bayes classifier on unigram features. Neural language model was used in Djuric et al. (2015). First they learn embedding of the text passages using paragraph2vec (Le and Mikolov, 2014). Then, they train a logistic regression classifier over those embeddings to classify into hateful and clean comments. Schmidt and Wiegand (2017) surveys the recent development in this field.

The first shared task on aggression identification (Kumar et al., 2018a) was held at the first workshop on Trolling, Aggression and Cyberbullying (TRAC). The goal was to classify social media posts into one of three labels (Overtly aggressive, Covertly aggressive, Non-aggressive).

The major contribution of the work can be summarized as a neural network based model which has LSTM units followed by an attention unit to embed the given social media post and training a classifier to detect aggression. We discuss our methods in section 2. Section 3 contains the details about the experiments and training data. In Section 4, we discuss the results and Section 5 concludes the paper with closing remarks.

2 Methodology

We hypothesize that aggression identification requires processing of the words of a sentence in a sequential manner. The positioning of a particular word at different places can alter the aggressiveness of the sentence. Example:

These aliens are filthy, but they live in a good neighbourhood. (Aggressive)
These aliens are good, but they live in a filthy neighbourhood. (Less aggressive)

Recurrent Neural Networks (Mikolov et al., 2010) are good at handling sequential data and have achieved good results in natural language processing tasks. As RNNs share parameters across time, they are capable of conditioning the model on all previous words of a sentence. Although theoretically it is correct that RNNs can retain information from all previous words of a sentence, but practically they fail at handling long-term dependencies. Also, RNNs are prone to the vanishing and exploding gradient problems when dealing with long sequences. Long Short-Term Memory networks (Hochreiter and Schmidhuber, 1997), a special kind of RNN architecture, were designed to address these problems.

2.1 Long Short-Term Memory networks

LSTMs use special units in addition to standard RNN units. These units include a 'memory cell' which can maintain its state for long periods of time. A set of non-linear gates is used to control when information enters the memory(Input gate), when it's outputted (Output gate), and when it's forgotten (Forget gate). The equations for the LSTM memory blocks are given as follows:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{5}$$

In these equations, x_t is the input vector to the LSTM unit, f_t is the forget gate's activation vector, i_t is the input gate's activation vector, o_t is the output gate's activation vector, h_t is the output vector of the LSTM unit and c_t is the cell state vector. w, u, B are the parameters of weight matrices and bias vectors which are learned during the training.

2.2 Attention

Here, the attention module is inspired by (Bahdanau et al., 2014). We deploy it after the LSTM unit. It helps the model decide the importance of each word for the classification task. It scales the representation of the words by a learned weighing factor, as determined by these equations:

$$e_t = h_t w_a \tag{6}$$

$$a_t = \frac{exp(e_t)}{\sum_{i=1}^{T} exp(e_i)} \tag{7}$$

$$v = \sum_{i=1}^{T} a_i h_i \tag{8}$$

In these equations, h_t is the hidden representation of a word at a time step t, w_a is the weight matrix for the attention layer, a_t is the attention score for the word at time t, and v is the final representation of the sentence obtained by taking a weighted summation over all time steps.

3 Experiments

3.1 Datasets

The training datasets for the English and Hindi sub-tasks are constructed by 11,999 and 12,000 Facebook posts and comments, respectively. The testing set includes 3,001 and 3,000 respectively. These are collected and manually annotated by the organizers. Most of the datum has an id and is classified into one of the three classes: OAG (Overtly aggressive), CAG (Covertly aggressive), NAG (Non-aggressive).

The distribution of the classes in the training dataset for English and Hindi are shown in Table 1. The organizers released a modified version of the data where they remove the rows without specific id. As the id is not important for prediction, we decided to work on the initially released dataset. The data collection methods used to compile the dataset for the shared task are described in Kumar et al. (2018b).

Class	Train (English)	Test (English)	Train (Hindi)	Test (Hindi)
Non-aggressive	5,051	1,233	2,275	538
Covertly aggressive	4,240	1,057	4,869	1,246
Overtly aggressive	2,708	711	4,856	1,217

Table 1: Class distribution in train and test sets

3.2 Preprocessing

Before feeding the Facebook comments to the LSTM classifier, we performed the following operations on the text:

1. We used the ekphrasis toolkit (Baziotis et al., 2017) for normalizing the occurrence of the following in the comments: URL, E-mail, percent, money, phone, user, time, date, and number. For example, URLs are replaced by <url>, and all occurrences of @someone are replaced by <user>.

2. We then passed the normalized text through the Social tokenizer. Unlike normal tokenizers, the Social tokenizer is specifically aimed at the unstructured social media content. It understands and parses complex emoticons, emojis and other unstructured expressions like dates, times, phone numbers etc.

3. Then, we removed the punctuations and used ekphrasis's inbuilt spell corrector on the text.

4. Lastly, we used NLTK's WordNet lemmatizer (Loper and Bird, 2002) to lemmatize the words to their roots.

3.3 Parameters

Our model uses an embedding layer of 100 dimensions to project each word into a vector space. We place a dropout (Srivastava et al., 2014) layer after this. To capture the context of the words passed from the dropout layer we use an LSTM layer having 100 hidden dimensions. As the LSTM cells already have non-linear activation functions, it helps the model capture non-linear semantics from the data. The output from the LSTM is then passed through an attention module. The attention module helps the model determine which word to give more importance to. The weighted output from attention module is passed through a fully-connected layer. To get the probabilities of each class, softmax function is applied to the output. We use the cross-entropy function to calculate the loss between the predicted and the target value. Adam optimizer is used with a learning rate of 0.001 to learn the weights of the model. The dropout rate was either 0.2 or 0.3 and is discussed in the Results section.

Although many machine learning classifiers like Naive Bayes, Decision Tree, Support Vector Machine or Random Forest could be used as a baseline classifier for this task. Due to constraint of time we have only used a Random Forest classifier. We train the classifier on a set of hand-crafted features. The features used are as follows:

1. Number of words with positive sentiment.

2. Number of words with negative sentiment.

3. Number of punctuations.

4. Total number of words.

5. Inverse of the 2nd feature.

6. Natural logarithm of the 2nd feature.

We use the lists made available by Hu and Liu (2004) for extracting positive and negative words.

4 Results

Due to a mistake on our side, we first submitted a model which considered only the first 45 words of the post/comment and used a dropout rate of 0.2, we denote this model as Eng-A in the tables. In Eng-B, we use dropout rate of 0.3 and considered all the words. RF baseline is the random forest classifier based model baseline of hand-crafted features.

System	F1 (weighted)
Random Baseline	0.3535
EF-A	0.5533
EF-B	**0.5746**

Table 2: Results for the English (Facebook) task.

System	F1 (weighted)
Random Baseline	0.3477
RF Baseline	0.3888
Eng-A	0.5304
Eng-B	**0.5548**

Table 3: Results for the English (Social Media) task.

For both the Hindi sub-tasks, we used the LSTM classifier with dropout probability of 0.3. We denote the model as Hi-A in the tables.

System	F1 (weighted)
Random Baseline	0.3571
Hi-A	**0.6032**

Table 4: Results for the Hindi (Facebook) task.

System	F1 (weighted)
Random Baseline	0.3206
Hi-A	**0.4703**

Table 5: Results for the Hindi (Social Media) task.

Looking at the confusion matrices of the English subtasks, it is clear that the model is performing well at classifying the non-aggressive comments from the aggressive or covertly aggressive comments. But it performs poorly and classifies a lot of over-aggressive and non-aggressive comments to covertly aggressive. The results in Malmasi and Zampieri (2018) also convey the same message.

5 Conclusion

In this paper, we present an LSTM network with an attention based classifier for aggression detection. It gives competitive results while relying only on the dataset provided. The performance reported in this paper could be further boosted by utilizing transfer learning methods from larger datasets, like using pre-trained word embeddings. Furthermore, the model tends to over-fit on the training data. Better generalization techniques, like the use of an increased dropout rate, might help in increasing the performance of the model.

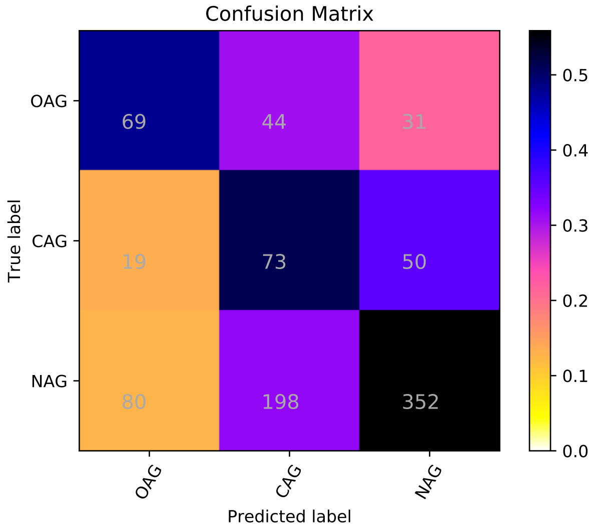

Figure 1: Confusion matrix for English (Facebook) task.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 29–30. International World Wide Web Conferences Steering Committee.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

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.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting Tweets Against Blacks. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January.