Detecting context abusiveness using hierarchical deep learning

Ju-Hyoung Lee
Yonsei University
Seoul, Republic of Korea
juhyounglee@yonsei.ac.kr

Jun-U Park
Yonsei University
Seoul, Republic of Korea
junupark@yonsei.ac.kr

Jeong-Won Cha
Changwon National University
Changwon, Republic of Korea
jcha@changwon.ac.kr

Yo-Sub Han
Yonsei University
Seoul, Republic of Korea
emmous@yonsei.ac.kr

Abstract

Abusive text is a serious problem in social media and causes many issues among users as the number of users and the content volume increase. There are several attempts for detecting or preventing abusive text effectively. One simple yet effective approach is to use an abusive lexicon and determine the existence of an abusive word in text. This approach works well even when an abusive word is obfuscated. On the other hand, it is still a challenging problem to determine abusiveness in a text having no explicit abusive words. Especially, it is hard to identify sarcasm or offensiveness in context without any abusive words. We tackle this problem using an ensemble deep learning model. Our model consists of two parts of extracting local features and global features, which are crucial for identifying implicit abusiveness in context level. We evaluate our model using three benchmark data. Our model outperforms all the previous models for detecting abusiveness in a text data without abusive words. Furthermore, we combine our model and an abusive lexicon method. The experimental results show that our model has at least 4% better performance compared with the previous approaches for identifying text abusiveness in case of with/without abusive words.

1 Introduction

As the number of social media data increases, abusive text such as online harassment, stalking, trolling and cyber-bullying becomes an important social issue. According to a Pew Research Center study[1] published in 2017, 66% of Internet users have observed someone being harassed and 41% have personally experienced harassment by themselves in online. There have been various attempts to detect or prevent abusive text and, in practice, the abusive word dictionary is the most efficient tool to identify abusive text even if an abusive word is obfuscated. However, if a text does not contain any abusive words explictly yet the abusiveness is clear in context, then it becomes a very challenging problem. For instance, E1) is an abusive comment that explicitly contains abusive words, and E2) is an abusive comment without abusive words.

- E1: Go you cocker cockuser motherfuck uncle suckefing you go fuck your mom you dirty little ass fuck bitch i will kill you i know where you live i will rape you yoru fucking ass.

- E2: I know how having the templates on their talk page helps you assert dominance over them. I know I would bow down to the almighty administrators. But then again, I'm going to go play outside... with your mon...

There are several approaches for detecting abusiveness using an abusive lexicon (Chen et al., 2012; Lee et al., 2018; Wiegand et al., 2018). These approaches work well when there is an abusive word in text. However, there is no explicit abusive words in text yet the text is abusive in context, the problem of identifying its abusiveness is challenging. We tackle this problem using an ensemble deep learning model.

[1] https://www.pewinternet.org/2017/10/10

Proceedings of the 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, pages 10–19
Hong Kong, China, November 4, 2019

Our model consists of two detection models. One is a Convolutional Neural Network (CNN) with bidirectional Long Short-Term Memory model (LSTM), and the other is the hierarchical C-LSTM model to understand the hierarchical structures in text. Each model specializes in understanding of long and short sentences. We evaluate our model using three popular benchmark social media datasets, Wikipedia, Facebook and Twitter. The experimental results show that our model outperforms the other baselines as well as the state of the art. We also run an additional experiment and evaluate the performance with respect to a sentence length for understanding context. The experimental results show that the hierarchical model is effective to solve the long dependency problem. Our contributions are summarized as follows:

- We design a hierarchical deep learning model that understands the hierarchical structure in long sentences with implicit abusiveness.
- We propose an ensemble model that combines two classifiers for understanding both of short and long sentences.
- We present an efficient abusive detection system using both our model and an abusive word dictionary.

We discuss the related work on abusiveness detection in Section 2 and propose our model in Section 3. We explain our datasets in Section 4. Then we evaluate our model by running several experiments in Section 5, and analyze the experimental results in Section 6. We suggest a few future directions and conclude the paper in Section 7.

2 Related Work

2.1 Text classification

Over the years, neural network models showed a great improvement in text classification. The emergence of Recurrent Neural Network (RNN) (Liu et al., 2016), which preserves the information continuity over time, and CNN (Kim, 2014), which preserves the local information of data, opened up a new indicator of text classification. Schwenk et al. (2017) presented Very-Deep CNN (VD-CNN) that uses only small convolutions and pooling operations for text processing. Zhou et al. (2015) proposed a C-LSTM model that combines CNN and LSTM to reflect the local information and the time continuity. Zhou et al. (2015) also introduced Attention-Based Bidirectional Long Short-Term Memory Networks (Attn-BLSTM) that can capture the semantic information among sentences using the attention mechanism (Bahdanau et al.). Researchers also added the structural characteristics of data into the learning model design. For example, Yang et al. (2016) proposed a hierarchical attention mechanism that mirrors the hierarchical structure of documents and solves the long-term dependency problem.

2.2 Lexicon-based abusive detection

As abusive text increases, there are several attempts to detect or prevent abusive text effectively. The most classical method is to determine the presence of abusive words. Chen et al. (2012) proposed the Lexical Syntactic Feature (LSF) architecture to detect offensive content and identify potential offensive users in social media together with user's writing style and cyberbullying content. Wiegand et al. (2018) proposed lexicons of abusive words that take advantage of a base lexicon by taking negative polar expressions. Lee et al. (2018) proposed a detection method by enhancing the abusive lexicon from the existing abusive words using Word2vec and deciding abusiveness together with n-grams and edit-distance for obfuscated abusive words.

2.3 Learning-based abusive detection

Djuric et al. (2015) proposed to learn the distributed low dimensional representation of comments using neural language models. Their model solves the high dimensionality and sparsity issues. Xiang et al. (2012) proposed a novel semi-supervised approach for detecting profanity content. It exploits linguistic regularities in profane language via statistical topic modeling. Zhang et al. (2016) noticed that lots of noise and errors in social media data made the abusive detection challenging. They proposed a Pronunciation-based Convolutional Neural Network (PCNN) and solved the error problem of data via phoneme codes of text as the features for a CNN. Zhang and Luo (2018) combined the convolutional and gated recurrent unit networks to detect hate speech on Twitter. They show that their method is able to capture both word sequence and order information in short texts compared to all the previous deep learning models. Srivastava et al. (2019) pre-

sented an approach that automatically classifies a toxic comment using a Multi Dimension Capsule Network. They also provide an analysis of their model's interpretation.

2.4 Ensemble model

Malmasi and Zampieri (2018) tackled the problem of identifying hate speech in social media using ensemble classifiers that consist of linear Support Vector Machine (SVM). Fauzi and Yuniarti (2018) suggested another ensemble method for an effective hate speech detection in Indonesian language and improved the detection performance. Cheng et al. (2019) utilized the time interval characteristic in social media for designing a detection model. In particular, they proposed a Hierarchical Attention Networks for Cyber-bullying Detection (ANCD) together with an ensemble technique applied to the deep learning model by separating users and messages from social media. It predicts the interval of time between two adjacent comments and shows that these tasks can improve the performance of cyber-bullying detection. van Aken et al. (2018) proposed an ensemble method that consists of Bidirectional LSTM (Bi-LSTM) and attention-based networks. They also conducted an in-depth error analysis of the toxic comment classification.

3 Methods and Ensemble

The proposed system consists of two parts as depicted in Figure 1. First, an abusive lexicon detects explicit abusiveness when there exists an (obfuscated) abusive word in text. Second, the ensemble deep learning model detects implicit abusiveness that does not contain any abusive words.

3.1 Lexicon of abusive words

We use an abusive lexicon (Wiegand et al., 2018) that takes advantage of the corpora and lexical resources. We also apply several efficient gadgets (Lee et al., 2018) based on blacklist, n-grams, punctuation and words with special characters to detect intentionally obfuscated words.

3.2 C-LSTM

Zhou et al. (2015) proposed C-LSTM that combines CNN and LSTM for text classification, and has advantages of both architectures. The CNN extracts a sequence of local information of sentences and LSTM obtains the representation of a sentence.

CNN: The CNN (Kim, 2014) extracts local information by preserving the word order and contextual information. We use the word embedding matrix W_e with 300 dimensions and convolution, which involves the 3 window vectors and 100 filters to obtain multiple features. We apply a non-linear function using a Rectified Linear Unit (ReLU) and the 1D max-pooling operation with pool size of 4 over the feature map to take the down-sampled maximum value. Let α_i denote d-dimensional word vectors through an embedding matrix W_e for the i^{th} word x_i in a sentence. We have a window vector w_i with k consecutive word vectors. A filter m convolves with the window vectors at each position in a valid way to generate a feature map c_i. For n filters with the same length, the generated n feature maps can be rearranged as feature representation for each window w_i as follow:

$$\begin{aligned}
\alpha_i &= W_e x_i, \\
w_i &= [\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+k-1}], \\
c_i &= f(w_i \circ m + b), \\
c_i &= ReLU(c_i), \\
\hat{c}_i &= max_4(c_i), \\
W &= [c_1, c_2, \ldots, c_n].
\end{aligned}$$

Bidirectional LSTM: The LSTM extracts orderly information (Zhang and Luo, 2018) by preserving a sequence of words or character n-grams. We use bidirectional LSTM, which has two LSTM layers instead of the standard LSTM to have information from backward and forward simultaneously. We use 100 features in the hidden state, followed by a dropout layer with a rate of 0.5. Afterward, we apply the 1D max-pooling operation to reduce the dimensionality of the LSTM output features $\overrightarrow{O}_j$ and $\overleftarrow{O}_j$. Finally, a linear-layer with the sigmoid function predicts the binary label classification and the softmax function predicts the multi-label classification.

$$\begin{aligned}
\overrightarrow{O}_j &= \overrightarrow{LSTM}(c_j), \\
\overleftarrow{O}_j &= \overleftarrow{LSTM}(c_j), \\
v &= max\{O\}, \\
p &= \{sigmoid, softmax\}(W_c v + b_c).
\end{aligned}$$

3.3 Hierarchical C-LSTM Networks

Yang et al. (2016) introduced hierarchical attention network for document classification that has

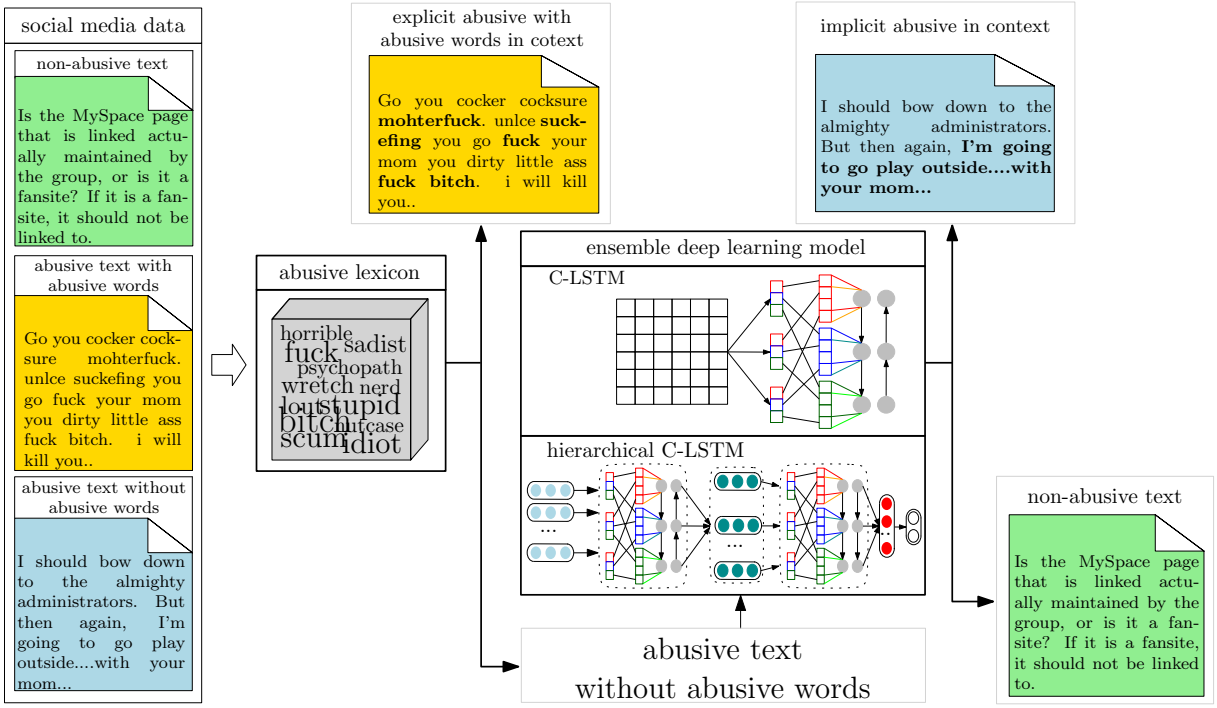

Figure 1: A proposed abusiveness detection mechanism by combining deep learning and an abusive lexicon

word attention and sentence attention. They suggested two distinctive characteristics: 1) it has a hierarchical structure that mirrors document has a hierarchical structure, and 2) it has two attention mechanism to prevent the loss of information in case of a long sentence. Since the abusiveness in context is preserved in a hierarchical structure, we propose a hierarchical C-LSTM network that is able to understand the hierarchical structure and uses a C-LSTM model instead of RNN attention model to extract the local information of a sentence. Let x_{it} be the t^{th} word vector in the i^{th} sentence s, and W_e be an embedding matrix.

$$X_{it} = W_e x_{it},$$
$$S_i = C_{LSTM}(X_{it}),$$
$$v = C_{LSTM}(S),$$
$$v = ReLU(v),$$
$$p = \{sigmoid, softmax\}(W_c v + b_c).$$

Hierarchical structure: A text often consists of several sentences and the structure of these multi-sentences is crucial to understand its context. We obtain the multi-sentence structure features using C-LSTM. Because online sentences often have punctuation errors including repeated occurrences, we split each sentence into fixed length in the data preprocessing described in Section 4.

3.4 Word Embedding

Word embedding provides a dense representation of words and their relative meanings. We use a pre-trained language model because there are many out of vocabulary words due to misspelling or newly created word. We use a fastText embedding (Bojanowski et al., 2017) of 300 dimensions trained with sub-word information on common crawl. For out-of-vocabulary words, we initialize the embedding with random weights.

3.5 Ensemble Learning

Each detection model has its own predictive power and scope. In the case of C-LSTM network, when a sentence is short, it can capture both word sequence and order information well. However, when a sentence is long, it cannot avoid the long-term dependency problem, which causes information loss. Hierarchical C-LSTM network can solve this problem to some extent by obtaining the local feature in each sentence. Therefore, we design an abusive detection model that is an ensemble of C-LSTM and hierarchical C-LSTM network as depicted in Figure 2. The proposed system also incorporates additional features associated with implicit abusiveness of text in local and global context level. For the ensemble, we concatenate the output of v_1 and v_2 through a C-LSTM and the output of u through a hierarchical C-LSTM. Then,

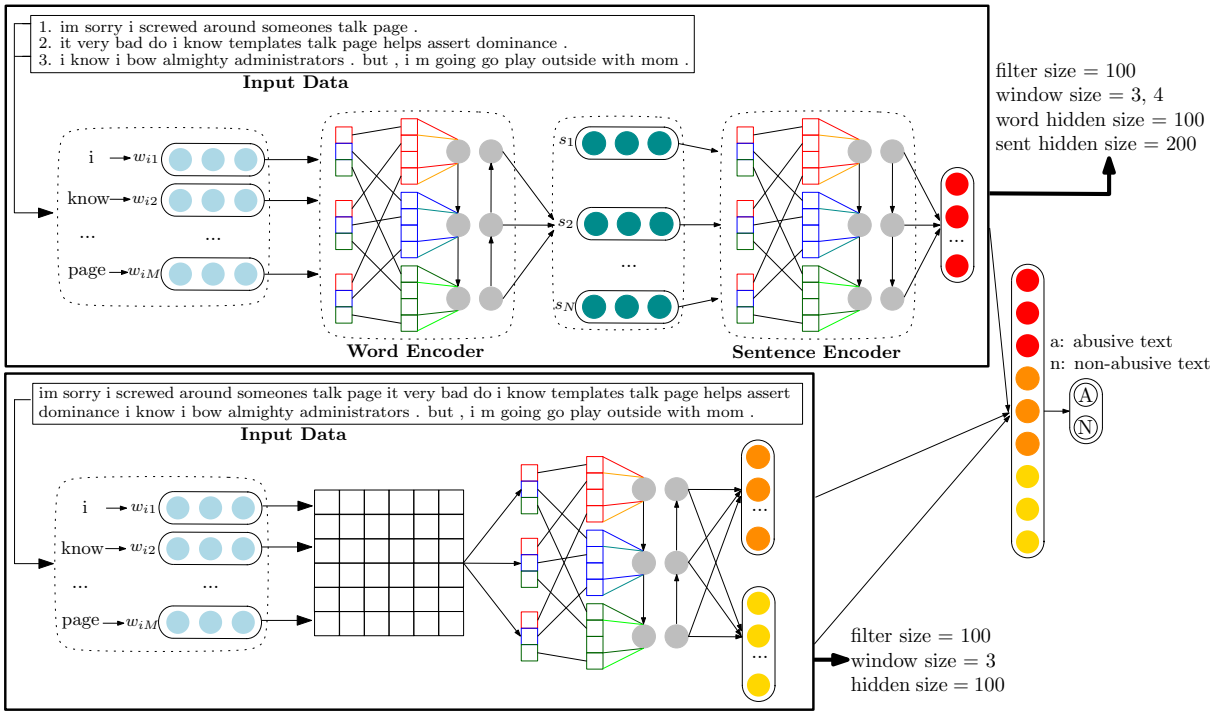

Figure 2: Ensemble of C-LSTM and hierarchical C-LSTM network

we apply a non-linear function using ReLU and feed this vector p to a fully-connected layer in order to predict the output.

$$v_1, v_2 = C_{LSTM(input)},$$
$$u = Hierarchical_{C_{LSTM(input)}},$$
$$p = concatenate(v_1, v_2, u),$$
$$p = ReLU(p),$$
$$p = linear_{layer(p)}.$$

4 Datasets

class	# of occurrences
Clean (Train)	80977 (96%)
Implicit Toxic (Train)	2948 (4%)
Clean (dev)	9019 (96%)
Implicit Toxic (dev)	307 (4%)
Clean (Test)	33541 (83%)
Explicit Toxic (Test)	5085 (13%)
Implicit Toxic (Test)	1158 (4%)

Table 1: Class distribution of Wikipedia dataset.

4.1 Kaggle Toxic Comment

Kaggle dataset is published by Google's Jigsaw for the toxic comment classification challenge. This dataset consists of comments from Wikipedia's talk page edits. Each comment categorized as one of the following six classes toxic, severe toxic, obscene, threat, insult and identity hate. We turn multi-class into binary-class to evaluate the performance of the abusive lexicon with ensemble deep learning model. We consider a toxic dataset if any of the six classes are applicable. Then, we split the dataset of 93,251 sentences

class	# of occurrences
NAG (Train)	4159 (46%)
Implicit CAG (Train)	3223 (36%)
Implicit OAG (Train)	1651 (18%)
NAG (Dev)	1029 (46%)
Implicit CAG (Dev)	806 (36%)
Implicit OAG (Dev)	420 (18%)
NAG (F)	491 (65%)
Explicit CAG (F)	35 (5%)
Explicit OAG (F)	56 (7%)
Implicit CAG (F)	95 (13%)
Implicit OAG (F)	73 (10%)
NAG (T)	431 (38%)
Explicit CAG (T)	85 (7%)
Explicit OAG (T)	103 (9%)
Implicit CAG (T)	328 (29%)
Implicit OAG (T)	188 (17%)

Table 2: Class distribution of Facebook (F) and Twitter (T) datasets.

into 90% training and 10% validation. We also use 39,784 test sentences provided by Kaggle as summarized in Table 1.

4.2 TRAC-1

TRAC-1 is a dataset shared by cyberbullying workshop. This dataset consists of 15,000 aggression annotated Facebook posts and comments. It makes a 3-way classification among Overtly Aggressive (OVG), Covertly Aggressive (CAG), and Non-Aggressive (NAG). We split the dataset into 80% training and 20% validation. Then, we use two test datasets from Facebook and Twitter provided by TRAC-1 to evaluate the performance as summarized in Table 2.

4.3 Data preprocessing

before preprocessing
I salute . . Neel Patel,, U r just amazing. Each & every comment of urs is true & correct...India n world need people like U...Love u my brother. God bless U...& pls don't stop here. Keep ur comments on every required post...
after preprocessing
i salute . ( )**(.)** neel patel **(,,)** u r just amazing. Each **(&)** every comment of urs is true **(&)** correct.**(..)**india n world need people like u.**(..)** love u my brother.god bless u. **(..)(&)** pls don**(')**t stop here. keep ur comments on every required post.**(..)**

Table 3: Data preprocessing example.

In the data preprocessing, we convert all characters to be lowercase, and remove whitespace, punctuations, non-English characters, URLs and Twitter and Facebook mentions. Table 3 is an example of this data preprocessing. We use a Natural Language Toolkit (NLTK) and regular expressions for data preprocessing.

5 Experiments

We run the following two experiments to verify the effectiveness of the deep learning module for implicit abusiveness and the abusive lexicon for explicit abusiveness:

1. Both training and testing datasets consist of implicit abusive text only.

2. The training dataset consists of implicit abusive text only, and the testing dataset consists of both explicit and implicit abusive text.

We use several baseline models and a few variants of our proposed ensemble model to evaluate the detection performance. We train all the models using cross-entropy as the loss function and Adam Optimizer (Kingma and Ba, 2015). For the evaluation metric, we choose the micro-average F1 measure because of the class strong imbalance in the dataset. In addition, we use Area under the Receiver Operating Characteristic curve (ROC AUC) to evaluate whether it can distinguish the difference between classes. All results are an average score of 5 evaluations.

5.1 Results

Deep learning performance: Table 4 compares our hierarchical model against the baselines as well as state-of-the-arts. Our model shows the best performance for the on Wikipedia dataset, however, there are no improvements from its baseline model C-LSTM and CNN for the Facebook and Twitter datasets. This is because the three datasets have different sentence lengths and sizes. The Wikipedia dataset has relatively large long sentences whereas the Facebook and Twitter datasets have rather short sentences. As mentioned in Section 3, since hierarchical C-LSTM applies hierarchical structure and often longer sentences preserve much more structural information, we have better performance on Wikipedia.

Ensemble performance: We use an ensemble of C-LSTM as a scalable approach to extract for small and short sentence features. Table 4 shows our ensemble with only one C-LSTM outperforms. Ensemble with two C-LSTM shows the better performance than individual models on three datasets. However, it has poor performance on Wikipedia compared to ensemble with only one C-LSTM. These show that the ensemble of additional models does not improve the performance.

Lexicon with deep learning performance: Our method combining an abusive lexicon and a deep learning model has the best performance. HAN improves performance of F1 measure 5.28% and AUC 7.06% on Wikipedia and our hierarchical model (HCL) improves performance of F1 measure 9.79% and ensemble model improves 12.74% on Facebook and Twitter. The result shows that the combined approach is more effective than any individual approach.

Comparison of F1 measure and AUC on three datasets consisting of **implicit abusive sentences**				
model	Wikipedia		Facebook	Twitter
	F1	AUC	F1	F1
LSTM (Wang et al., 2015)	94.24	91.95	50.08	50.17
Bi-LSTM (Zhou et al., 2016)	95.55	91.91	50.93	50.50
CNN (Kim, 2014)	95.46	90.95	53.83	60.50
C-LSTM (Zhang et al., 2018)	95.70	91.66	52.88	59.60
HAN (Yang et al., 2016)	96.32	89.21	50.25	54.09
HCL	**96.36**	92.91	53.15	58.43
HCL+C-LSTM	96.08	**93.03**	**54.77**	60.55
HCL+C-LSTM+C-LSTM	95.61	93.00	54.12	**62.51**
Comparison of F1 measure and AUC on three datasets consisting of **explicit and implicit abusive sentences**				
model	Wikipedia		Facebook	Twitter
	F1	AUC	F1	F1
LSTM	90.35	92.02	53.88	53.71
Bi-LSTM	91.65	91.94	54.74	52.80
CNN	91.45	92.06	53.54	56.33
C-LSTM	91.67	92.13	53.74	57.23
HAN	91.53	90.93	51.97	55.99
HCL	91.89	92.31	51.13	53.22
HCL+C-LSTM	91.54	92.55	53.91	52.62
HCL+C-LSTM+C-LSTM	**91.97**	**92.71**	**55.11**	**57.50**
Comparison of F1 measure and AUC on three datasets consisting of **explicit and implicit abusive using both an abusive lexicon and a deep learning model**				
model	Wikipedia		Facebook	Twitter
	F1	AUC	F1	F1
LSTM	94.97	98.50	58.36	56.26
Bi-LSTM	96.12	98.32	59.07	56.54
CNN	96.04	98.31	61.48	65.53
C-LSTM	96.25	98.45	60.69	64.54
HAN	96.81	97.99	58.37	59.57
HCL	**96.82**	98.68	60.92	63.51
HCL+C-LSTM	96.58	**98.73**	60.74	65.36
HCL+C-LSTM+C-LSTM	96.17	98.70	**62.51**	**67.09**

Table 4: Results of different models on Wikipedia, Facebook and Twitter datasets, HAN: Hierarchical Attention Neural Net, HCL: Hierarchical C-LSTM. Explicit abusive is when there is an (obfuscated) abusive word, and implicit abusive is no abusive word yet abusive in context.

model	data length (|T|)	F1	AUC
C-LSTM	$0 <\|T\|< 100$	**87.61**	94.36
HCL		84.57	93.88
HCL+C-LSTM+C-LSTM		87.41	**94.62**
C-LSTM	$100 <\|T\|< 200$	87.45	94.22
HCL		86.35	93.90
HCL+C-LSTM+C-LSTM		**87.87**	**94.61**
C-LSTM	$200 <\|T\|$	87.33	95.34
HCL		90.52	**96.56**
HCL+C-LSTM+C-LSTM		**90.69**	96.55

Table 5: Comparisons of F1 measure and AUC using Wikipedia dataset which has under 100 words, over 100 and under 200 words and over 200 words.

type	**data**		**example**	**predicted label**	**true label**
type-1	W	52%	black people deserve to die	0	1
	F	37%	shutdownjnu do it	NAG	CAG
	T	39%	families as well lol	NAG	CAG
type-2	W	13%	actually help us u dont help at all so help dick heads from ur unexpected guest	0	1
	F	3%	what to say about shutdownjnu just shut up	NAG	OAG
	T	2%	sensex in maternity ward .good morning cnbc tv	NAG	CAG
type-3	W	35%	poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo poo	1	0
	F	60%	send umar khalid to afzal before its too late . do n t let him become another afzal guru in real ! shutdownjnu madarsajnu	NAG	OAG
	T	59%	kinner to vo h jo bar bar pakistan dwara hamare uper hamle hone par bhi apna much chhipa k baitha h kinner vo h jo bar bar bina bulaye pakistan ja raha h	CAG	NAG

Table 6: Example of error caused by the proposed model for the Wikipedia (W), Facebook (F) and Twitter (T) dataset. 0: abusive, 1: non-abusive.

5.2 Experiment with short/long sentences

Previously, we have claimed that C-LSTM is specialized in short sentences and Hierarchical C-LSTM is specialized in long sentences. We verify this claim using different sentence lengths. Because the Wikipedia dataset contains a lot of punctuation errors, we use the number of words instead of sentences for this experiment. We categorize the dataset into three parts (<100, 100∼200, 200<) by counting the number of words in sentences. Note that Founta et al. (2019) suggests a sentence of under 100 words to be a short sentence. We use the micro-average F1 measure and AUC for evaluation metric. As presented in Table 5, C-LSTM has better performance of F1 measure 3.04%, 1.1% and AUC 0.48%, 0.32% compared to HCL in sentences with under 100 words, and over 100 and under 200 words. HCL has better performance of F1 measure 3.19% and AUC 1.22% compared to C-LSTM in sentences which have over 200 words. This experiment verifies that each model is specialized in relation to the size of sentences. Finally, our ensemble model shows the best performance except for a slight difference of 0.2% compared to C-LSTM.

6 Error analysis

We manually categorize the resulting errors into three types: short sentences of less than five words (type-1), sentences with new explicit words (type-2), and sentences having misspelled words, which cause wrong decisions (type-3). Table 6 shows examples of these types of errors.

From the type-1, we can see that our model is confused in understanding the meaning of short sentences of less than five words. It is hard for our model to understand the context of short sentences, since these are few words that does not contain abusive words. The type-2 is an error caused by obfuscated and new abusive words that are not in the current abusive lexicon, such as "esss", "a**hole", "betches", and "b1tch". In order to solve these issues, we need to improve and modify the abusive lexicon furthermore. The type-3 is an error caused by the presence of repetitive and misspelled words. Because online comments often do not basically follow formal language conventions, there are many unstructured, informal and often misspelled and abbreviations. These make the abusive detection very difficult. One can handle these problems in two ways: preprocessing the data with grammar checker or improving the performance with pre-trained embedding model.

7 Conclusion and Future work

We have tackled the problem of detecting abusiveness when there are no abusive words in text using deep learning. We have designed a hierarchical deep learning model that extracts global features for long sentences. We have also proposed an ensemble models that combine two classifiers extracting local and global features. Finally, we have combined our model for context abusiveness and an abusive lexicon method. We have evaluated the proposed system on Wikipedia, Facebook and Twitter datasets. The experimental results confirm that our hierarchical model outperforms in implicit abusive sentences of more than 100 words. Ensemble model outperforms baselines as well as the state of the art in most cases. The combination of an abusive lexicon and a deep learning model shows the best performance in comparison to the individual method.

We plan to develop methods to detect implicit abusiveness in short sentences. Furthermore, we aim to build a new abusive detection method using additional language models.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00247).

References

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online*, pages 33–42.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *Proceedings of the International Conference on Privacy, Security, Risk and Trust, and International Confernece on Social Computing*, pages 71–80.

Lu Cheng, Ruocheng Guo, Yasin Silva, Deborah Hall, and Huan Liu. 2019. Hierarchical attention networks for cyberbullying detection on the instagram social network. In *Proceedings of the Society for Industrial and Applied Mathematics International Conference on Data Mining*, pages 235–243.

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.

Mochammed Ali Fauzi and Anny Yuniarti. 2018. Ensemble method for indonesian twitter hate speech detection. *Electrical Enginerring and Computer Science*, 11:294–299.

Antigoni-Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2019. A unified deep learning architecture for abuse detection. In *Proceedings of the 11th Conference on Web Science*, pages 105–114.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Ho-Suk Lee, Hong-Rae Lee, Jun-U. Park, and Yo-Sub Han. 2018. An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems*, 113:22–31.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2873–2879.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Experimental and Theoretical Artificial Intelligence*, 30(2):187–202.

Holger Schwenk, Loïc Barrault, Alexis Conneau, and Yann LeCun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1107–1116.

Saurabh Srivastava, Prerna Khurana, and Vartika Tewari. 2019. Detecting aggression and toxicity in comments using capsule network. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 157–162.

Xin Wang, Yuanchao Liu, Chengjie Sun, Baoxun Wang, and Xiaolong Wang. 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1343–1353.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. 2018. Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1046–1056.

Guang Xiang, Bin Fan, Ling Wang, Jason I. Hong, and Carolyn Penstein Rosé. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st International Conference on Information and Knowledge Management*, pages 1980–1984.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Xiang Zhang, Jonathan Tong, Nishant Vishwamitra, Elizabeth Whittaker, Joseph P. Mazer, Robin Kowalski, Hongxin Hu, Feng Luo, Jamie Macbeth, and Edward Dillon. 2016. Cyberbullying detection with a pronunciation based convolutional neural network. In *Proceedings of the 15th International Conference on Machine Learning and Applications*, pages 740–745.

Ziqi Zhang and Lei Luo. 2018. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *The Computing Research Repository, CoRR*, abs/1803.03662.

Ziqi Zhang, David Robinson, and Jonathan A. Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *Proceedings of the Semantic Web – 15th International Conference*, pages 745–760.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM neural network for text classification. *The Computing Research Repository, CoRR*, abs/1511.08630.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. In *Proceedings the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3485–3495.