YNU_oxz at SemEval-2020 Task 12: Bidirectional GRU with Capsule for Identifying Multilingual Offensive Language

Xiaozhi Ou, Hongling Li*
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
*Corresponding author, honglingli66@126.com

Abstract

This article describes the system submitted to SemEval-2020 Task 12 OffensEval 2: Multilingual Offensive Language Recognition in Social Media. The task is to classify offensive language in social media. The shared task contains five languages (English, Greek, Arabic, Danish, and Turkish) and three subtasks. We only participated in subtask A of English to identify offensive language. To solve this task, we proposed a system based on a Bidirectional Gated Recurrent Unit (Bi-GRU) with a Capsule model. Finally, we used the K-fold approach for ensemble. Our model achieved a Macro-average F1 score of 0.90969 (ranked 27/85) in subtask A.

1 Introduction

Offensive language is ubiquitous in social media, and individuals often uses the anonymity of computer communications for some anti-social network behaviors, including cyberbullying (Xu et al., 2012), malicious provocation (Kwok and Wang, 2013), and offensive language (Cheng et al., 2017). The widespread dissemination of offensive content in social media is a cause of concern for governments and many technology companies around the world. One of the most common and effective strategies for solving offensive language problems on the network is to train systems that can recognize such content.

SemEval-2020 OffensEval 2 is proposed for multilingual offensive language recognition in social media (Zampieri et al., 2020). The shared task contains three subtasks and five languages (English, Greek, Arabic, Danish, and Turkish), where subtask A is a coarse-grained binary classification, which aims to the identification of offensive language. Participating systems need to divide Tweet into two categories: Offensive (OFF) and Not Offensive (NOT). In this competition, we only participated in subtask A of the English language. We used deep learning to build a bidirectional GRU(Bi-GRU) with Capsule model (Yang et al., 2018), among them, GRU is simpler and more efficient than the traditional LSTM model (Chung et al., 2014). Our model used bidirectional GRU (Bi-GRU) (Bahdanau et al., 2014) to process the sequence from two directions, utilizing both the previous and future context, and capsule is a group of neurons that use vectors to represent parameters, capsule network uses the inner product method to cluster the input features.

The rest of this article is organized as follows. Section 2 introduces related work on multilingual offensive language identification. Section 3 describes the models and data. Section 4 presents the experimental results. Finally, we summarize in Section 5.

2 Related Work

In recent years, offensive language has prevailed on social media, and social media has become the most popular media among users. According to the survey (QEV Analytics and of America, 2009), it was observed that 70% of adolescents used social media sites every day, and users shared their opinions

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2251–2257
Barcelona, Spain (Online), December 12, 2020.

through social media such as Twitter, Facebook, Microblog, etc. Kumar et al. (2018) attempted to identify hate speech. On the one hand, users benefit from social media by learning or interacting with other users; on the other hand, they face offensive online content. In light of more recent survey of hate speech and offensive language detection, we recommend Schmidt and Wiegand (2017) and Fortuna and Nunes (2018). Schmidt and Wiegand (2017) investigated features widely used for hate speech detection, including simple surface features, word generalization, 88 knowledge-based features, etc. Fortuna and Nunes (2018) believed that the field of automatic detection of hate speech and offensive language in text is very important for online social platforms and has unquestionable potential for social impact. Davidson et al. (2017) presented the results of hate speech detection using word n-grams and emotional vocabulary, and provided insights into examples of misclassification.

In addition to recently published research, a number of related sharing tasks have been organized. Among them, Gemeval2018 (Wiegand et al., 2018) is about offensive language recognition and aims to promote research on offensive content recognition in German language microblogs. The best team's system is to train three basic classifiers (maximum entropy and two random forest sets) using five disjoint feature sets, and then used the maximum entropy element-level classifier for final classification (Montani, 2018). In the SemEval-2019 shared tasks HatEval (Basile et al., 2019) and OffensEval (Zampieri et al., 2019b), HatEval is a multilingual detection of hate speech against immigrants and women on Twitter. Fermi team is the best team of Hateval. It proposes a SVM model with RBF kernel and uses sentence embedding in Google general sentence encoder as a function (Indurthi et al., 2019). OffensEval is about the identification and classification of offensive language in social media. The NULI team is the best performing team, they use BERT-base without default parameters (Liu et al., 2019). HASOC2019 (Mandl et al., 2019) is proposed to identify hate speech and offensive content in Indo-European languages. Its purpose is to develop powerful technologies capable of processing multilingual data and to develop a transfer learning method that can utilize cross-lingual data. The optimal system is a system based on ordered neuron LSTM (ON-LSTM) and attention model and adopts K-folding approach for ensemble (Wang et al., 2019).

3 Methodology and Data

3.1 Data description

We only participated in English subtask A. The official English dataset provided this year is different from the Offensive Language Identification DataSet (OLID) (Zampieri et al., 2019a). The format of the dataset instance is as follows:

$$\text{ID}\ \langle TAB \rangle \quad \text{TWEET}\ \langle TAB \rangle \quad \text{AVG\_CONF}\ \langle TAB \rangle \quad \text{CONF\_STD}$$

where AVG_CONF is the average of the confidences predicted by several supervised models for a specific instance to belong to the positive class for that subtask. The positive class is OFF for subtask A. CONF_STD is the confidences' standard deviation from AVG_CONF for a particular instance. For official provided English datasets containing scores rather than labels, the scores are confidence measures produced by unsupervised learning methods. We used a 0.5 average confidence threshold (AVG_CONF) to map the scores to the OffensEval labels. Based on the principle that the more the number of training sets, the better the performance of the model may be. We randomly selected the maximum 100,000 pieces of data that our experimental equipment can accommodate as the training set. And based on the number of OffensEval 2019 test sets, we randomly selected 3887 data consistent with the number as the validation set data required for this experiment.

3.2 Data preprocessing

We performed some operations to pre-process the data, Tweets was processed using the Tweetokenize tool [1]. We used Emoji substitution and HashTag segmentation and all "@use" is replaced with username, and the frequency of consecutive "@USER" is limited to three times to reduce redundancy. We also removed punctuation, replaced all uppercase letters with lowercase letters, restored abbreviations, etc.

[1] https://www.github.com/jaredks/tweetokenize

3.3 Bi-GRU with a Capsule model

Our proposed network architecture is shown in Figure 1. Our model is built on Bi-GRU with Capsule, where GRU is a variant of LSTM. Next, we briefly describe the details of the system.

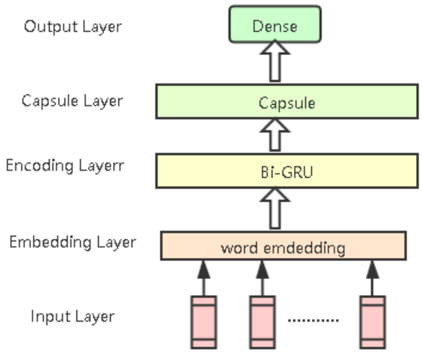

Figure 1: The architecture of Bi-GRU with a Capsule model for English Subtask A

- **Input layer:** This layer mainly inputs all pre-processed text data into the model.
- **Embedding layer:** The embedding layer converts words in an existing dictionary input through a pre-trained word vector model into vectors.
- **Encoding layer:** Chung et al. (2014) proposed an LSTM variant called gate recursive unit (GRU), GRU is to combine the forget gate and input gate in LSTM into update gate. It makes GRU simpler and more efficient than traditional LSTM models (Wang et al., 2018). In the encoding layer, we used the structure of a bidirectional GRU to encode vectorized text to establish this contextual connection. Bi-GRU is a neural network model consisting of unidirectional GRU with opposite directions and whose output is jointly determined by the states of these two GRUs. The input of the forward GRU is the forward sequence of the input of the previous layer, and the input of the backward GRU is the reverse sequence of the input of the previous layer. At each moment, the input provided two GRUs with directions opposite at the same time, and the output is decided by the two unidirectional GRUs jointly. The current hidden layer state of Bi-GRU is jointly determined by three parts: current input x_t, output $\overrightarrow{h_{t-1}}$ of forward hidden layer state and output $\overleftarrow{h_{t-1}}$ of backward hidden layer state at *t-1* moment:

  $$\overrightarrow{h_t} = GRU(x_t, \overrightarrow{h_{t-1}}) \tag{1}$$

  $$\overleftarrow{h_t} = GRU(x_t, \overleftarrow{h_{t-1}}) \tag{2}$$

  $$h_t = [\overrightarrow{h_t} : \overleftarrow{h_t}] \tag{3}$$

  where the GRU() function represents a non-linear transformation of the input word vector, and encodes the word vector into the corresponding GRU hidden layer state. $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ respectively represent the forward hidden state and the backward hidden state corresponding to the bidirectional GRU at *t* moment; h_t express the vector that contact $\overrightarrow{h_t}$ with $\overleftarrow{h_t}$.
- **Capsule layer:** In the deep learning model, spatial patterns are aggregated at a lower level, which helps to represent higher-level concepts. We used the Capsule network to enhance the model's feature extraction capabilities, spatial insensitivity methods are inevitably limited by the abundant text structure (such as saving the location of words, semantic information, grammatical structure, etc.), difficult to effectively encode and lack of text expression ability. The Capsule network effectively improved this disadvantage by using neuron vectors instead of individual neuron nodes of

traditional neural networks to train this new neural network in the dynamic routing way. Capsule's parameter update algorithm is routing-by-agreement (Sabour et al., 2017), a lower-level capsule prefers to send its output to higher-level capsules whose activity vectors have a big scalar product with the prediction coming from the lower-level capsule. The calculation formula of Capsule is as follows:

$$V_j = \frac{\| S_j \|^2}{1 + \| S_j \|^2} \frac{S_j}{\| S_j \|} \tag{4}$$

$$S_j = \sum_i C_{ij} \hat{u}_{j|i}, \qquad \hat{u}_{j|i} = W_{ij} u_i \tag{5}$$

where V_j is the vector output of capsule j and S_j is its total input, prediction vectors $\hat{u}_{j|i}$ is by multiplying the output u_i of a capsule in the layer below by a weight matrix W_{ij}, the C_{ij} are coupling coefficients that are determined by the iterative dynamic routing process.

- **Output layer:** This layer classifies and predicts the final aggregated information.

3.4 K-folding ensemble

In this paper, in order to enhance the overall classification performance of the model, we used a K-fold ensemble method. The design idea of this method comes from K-fold cross-validation, we randomly divided the source data into K parts and used the K-1 subsets to do the training, the remaining subset is the validation set, and then this process is repeated K times. Finally, the K results are subjected to an accumulation averaging operation to obtain the final output. The purpose of performing the K-fold ensemble is to train to different data sets during each fold training process, and to extract different features during the model feature extraction process, which can further improve the generalization ability of the model.

4 Experiment and results

4.1 Ablation experiment

System	Dataset	Accuracy(Validation set)
ON-LSTM+Attention [S1]	OLID	76.93%
RnnCnn_model(Bi-GRU+Conv1D) [S2]	OLID	79.71%
ON-LSTM+Attention [S1]	100000 unlabeled dataset	94.28%
RnnCnn_model(Bi-GRU+Conv1D)[S2]	100000 unlabeled dataset	96.53%
Bi-LSTM+Attention [S3]	100000 unlabeled dataset	96.02%
Bi-LSTM+Capsule [S4]	100000 unlabeled dataset	96.77%
Bi-GRU+Attention [S5]	100000 unlabeled dataset	97.10%
Bi-GRU+Capsule [S6]	100000 unlabeled dataset	98.86%

Table 1: Performance ablation experiments on the validation set to compare models

For the unlabeled English dataset released this year, we consider whether to introduce the OLID dataset provided by OffensEval 2019. Therefore, on the same validation set, we conducted a comparative experiment on the selection of the training set. The relevant description of the data set used in this section is as described in section 3.1. As shown in Table 1, we conducted experiments on these two data sets on the same model. It can be observed that compared with the OLID data set, the randomly selected 100,000 unlabeled datasets improved on systems S1 and S2, respectively. It is 17.35% and 16.82%. Therefore, we selected randomly selected 100,000 unlabeled data sets as the training set for this experiment. On this basis, we conducted ablation experiments on the system to verify the performance of the model. Observing the change from system S2 to S6 in Table 1, we can find that system S6 reached 98.86%, which is an increase of 2.33% over system S2.

4.2 Experiment setting

In our model, the pre-trained word embedding we used is FastText [2], which is provided by Mikolov et al. (2017). It is a 2 million word vector trained using subword information on Common Crawl with 600B tokens, and it's dimension is 300(crawl-300d-2M.vec). In the encoding layer, we set the hidden units to 32. In the Capsule layer, we set num_capsule = 10, dim_capsule = 16, routings = 4. The Flatten layer is connected behind the Capsule layer, this is to turn multidimensional input into one-dimensional, so as to achieve the transition from a convolution layer to a full connection layer. A layer of Dense with Relu activation function is connected behind the Flatten layer, and the number of hidden units is 16. We then added the Dropout layer and the BatchNormalization layer, with dropout is set to 0.5. In the output layer, we used the sigmoid activation function for binary classification. The loss function of this model is binary cross-entropy, and the optimizer is adam. we set the batch size to 64 and the epoch to 20 for training. Finally, we used the K-fold method for ensemble, and K is set to 5.

4.3 Result

This English subtask A evaluates the classification system by calculating a macro-averaged F1-score, which takes into account both the precision and recall of the classification model. The F1 score can be regarded as a harmonic average of the model's precision and recall. Its maximum value is 1 and its minimum value is 0. Macro-averaging is to first statistical index value for each class and then calculates the arithmetic average for all classes. Table 2 showed the official results of the English subtask A. The confusion matrix of Bi-GRU with a Capsule model for the English subtask A is shown in Appendix A.

System	**F1(Macro-averaging)**
All NOT baseline	0.41933 (baseline)
The best team	0.92226 (1)
Our model	0.90969 (27)

Table 2: Results for English Sub-task A

5 Conclusion

This year we participated in the English subtask A for multilingual offensive language recognition, which is automatically classifying offensive language in social media. This paper proposed a model for English offensive language recognition. We used the Bi-GRU model for classification, and used the Capsule network to improve the feature extraction capability of the model. Although our model overall performance is not the best, preliminary results indicate what we should do next. In future research, we consider the impact of unlabeled datasets on the results, we will consider the introduction of transfer learning, and how to optimize the parameters is also a very important issue. We will also consider offensive language recognition in other languages (Greek, Arabic, Spanish and Danish).

References

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

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Justin Cheng, Michael S. Bernstein, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. In *Computer Supported Cooperative Work*, volume abs/1702.01119.

[2] https://fasttext.cc/docs/en/english-vectors

Junyoung Chung, aglar Glehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. abs/1412.3555.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Paula Fortuna and Srgio Nunes. 2018. A survey on automatic detection of hate speech in text. 51:85:1–85:30.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Nikhil Chakravartula, Manish Gupta, and Vasudeva Varma. 2019. FERMI at SemEval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 70–74, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Ritesh Kumar, Guggilla Bhanodai, Rajendra Pamula, and Maheshwar Reddy Chennuru. 2018. Trac-1 shared task on aggression identification: Iit (ism)@ coling18. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 58–65.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. *AAAI Conference on Artificial Intelligence*.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 14–17.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *CoRR*, abs/1712.09405.

Joaquın Padilla Montani. 2018. Tuwienkbs at germeval 2018: German abusive tweet detection. In *14th Conference on Natural Language Processing KONVENS*, volume 2018, page 45.

Ltd. QEV Analytics and United States of America. 2009. National survey of american attitudes on substance abuse xiv: Teens and parents.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules.

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*, pages 1–10.

Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual lstm model for sentiment intensity prediction. *Neurocomputing*, 322:93–101.

Bin Wang, Yunxia Ding, Shengyan Liu, and Xiaobing Zhou. 2019. Ynu_wb at HASOC 2019: Ordered neurons LSTM with attention for identifying hate speech and offensive language. In Parth Mehta, Paolo Rosso, Prasenjit Majumder, and Mandar Mitra, editors, *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, volume 2517 of *CEUR Workshop Proceedings*, pages 191–198. CEUR-WS.org.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. 09.

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*.

Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating capsule networks with dynamic routing for text classification. In *Conference on Empirical Methods in Natural Language Processing*, pages 3110–3119.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

A Appendix A : Confusion matrix of Bi-GRU with Capsule model for English subtask A

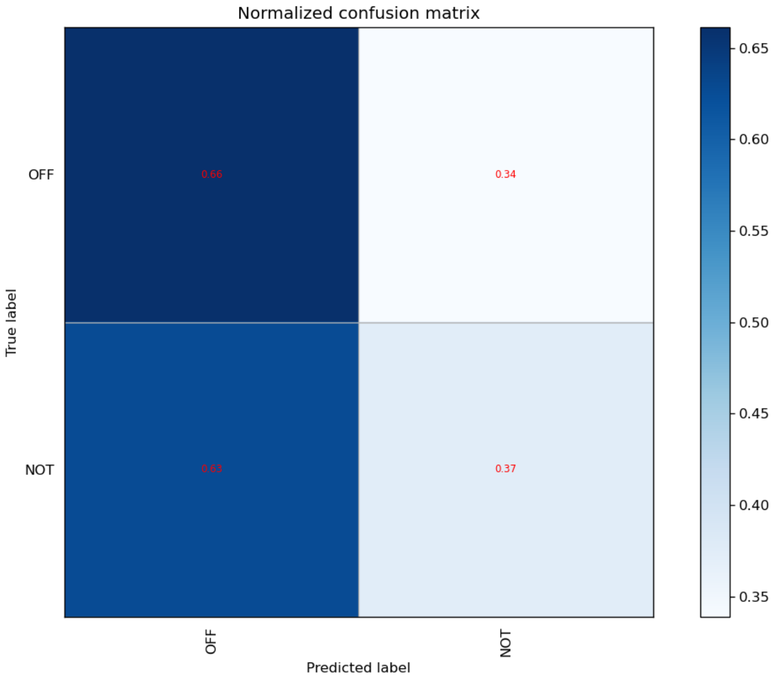

Figure 2: Confusion matrix of Bi-GRU with Capsule model for English subtask A. As can be seen from the confusion matrix, our model has a lot of NOT prediction errors into OFF, the reason for this may be that the data in the training data set is not too balanced.