YNU-HPCC at SemEval-2021 Task 6: Combining ALBERT and Text-CNN for Persuasion Detection in Texts and Images

Xingyu Zhu, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
`Contact: xyzhu@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn`

Abstract

In recent years, memes combining image and text have been widely used in social media, and memes are one of the most popular types of content used in online disinformation campaigns. In this paper, our study on the detection of persuasion techniques in texts and images in SemEval-2021 Task 6 is summarized. For propaganda technology detection in text, we propose a combination model of both ALBERT and Text-CNN for text classification, as well as a BERT-based multi-task sequence labeling model for propaganda technology coverage span detection. For the meme classification task involved in text understanding and visual feature extraction, we designed a parallel channel model divided into text and image channels. Our method[1] achieved a good performance on subtasks 1 and 3. The micro F_1-scores of 0.492, 0.091, and 0.446 achieved on the test sets of the three subtasks ranked 12th, 7th, and 11th, respectively, and all are higher than the baseline model.

1 Introduction

The intentional shaping of information to promote a predetermined agenda is called propaganda. Propaganda uses psychological and rhetorical techniques to achieve its purpose. Propaganda techniques generally include the use of logical fallacies and appeal to the emotions of the audience. In recent years, memes combining images and text have been widely used in social media, and the use of memes can easily and effectively attract a large number of users on social platforms. Memes are one of the most popular types of content used in online disinformation campaigns (Martino et al., 2020) , and memes applied in a disinformation campaign achieve their purpose of influencing users through rhetorical and psychological techniques. Therefore, it is meaningful to research computational techniques for automatically detecting propaganda in particular content.

The SemEval 2021 Task 6 (Dimitrov et al., 2021) consists of three subtasks:

- Subtask 1 - Given only the "textual content" of a meme, identify which of the 20 techniques are used. The 20 techniques include appeal to authority, loaded language, and name calling or labeling.
- Subtask 2: Given only the "textual content" of a meme, identify which of the 20 techniques are used along with the span(s) of the text covered by each technique.
- Subtask 3: Given a meme, identify which of the 22 techniques are used for both the textual and visual content of the meme. These 22 technologies include the 20 technologies in subtasks 1 and 2, and 2 technologies, i.e., transfer and appeal to (strong) emotions, are added.

The detection of propaganda techniques in texts is similar to a text sentiment analysis, and both can be attributed to text classification tasks. In a previous study, Peng et al. (2020) used the adversarial learning of sentiment word representations for a sentiment analysis. A tree-structured regional CNN-LSTM (Wang et al., 2020) and dynamic routing in a tree-structured LSTM (Wang et al., 2019) were used for a dimensional sentiment analysis. In previous SemEval competitions, Dao et al. (2020) used GloVe-LSTM and BERT-LSTM models, and Paraschiv et al. (2020) used an ensemble model containing BERT and BiLSTM to detect both spans and categories of propaganda techniques in news articles (Da San Martino et al., 2020) . In addition, in multimodal analysis combining images and

[1] The code of this paper is available at: `https://github.com/zxyqujing/SemEval-2021-task6`

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1045–1050
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

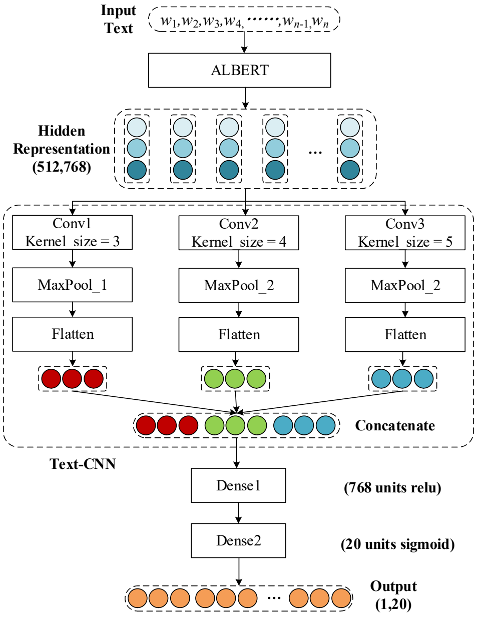

Figure 1: ALBERT-Text-CNN model architecture.

text, Yuan et al. (2020) proposed a parallel channel ensemble model combining BERT embedding, BiLSTM, attention and CNN, and ResNet for a sentiment analysis of memes. Li et al. (2019) proposed a Visual BERT model that aligns and fuses text and image information using transformers (Vaswani et al., 2017) .

In this paper, we propose three different systems for the three subtasks in SemEval-2021 Task 6. For subtask 1, we added a Text-CNN layer after the pre-trained model ALBERT to fine-tune it for a multi-label classification of text. For subtask 2, we used the idea of partitioning to transform the problem into the detection of 20 techniques for each text separately. BERT was used in the model for text feature extraction followed by multi-task sequence labeling, and the results of each task were combined to obtain the final results. For subtask 3, we built the system using a parallel channel model containing text and image channels. The text channel used both the ALBERT and Text-CNN models to extract features of text in the meme, and the image channel used ResNet and VGGNet for image feature extraction. The information extracted by the two parallel channels was then combined through a fully connected layer after concatenation. Using micro F_1-scores as metrics, the results of the proposed model in subtasks 1, 2, and 3 were 0.625, 0.215, and 0.636, respectively, on the dev set.

The remainder of this paper is organized as follows. First, section 2 describes the details of the ALBERT and Text-CNN used in our system. Section 3 then presents the experimental results. Finally, some concluding remarks are presented in section 4.

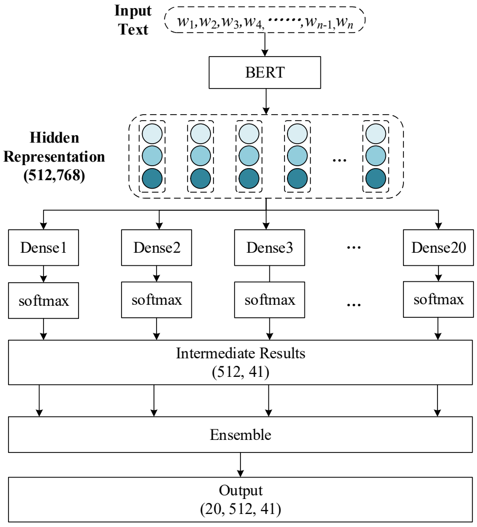
Input Text
$w_1, w_2, w_3, w_4, \cdots\cdots, w_{n-1}, w_n$
BERT
Hidden Representation (512,768)
Dense1
Dense2
Dense3
Dense20
softmax
softmax
softmax
softmax
Intermediate Results (512, 41)
Ensemble
Output (20, 512, 41)

Figure 2: Architecture of multi-task sequence labeling model.

2 System Overview

2.1 Subtask 1

Subtask 1 requires a detection model that uses only the textual features of the meme content and detects which of the 20 propaganda techniques were used. This is a multi-label classification problem for text, based on the pre-trained ALBERT model and added a Text-CNN layer. As illustrated in Figure 2, the proposed model includes an ALBERT layer, a Text-CNN layer, a fully connected layer, and an output layer.

- ALBERT (Lan et al., 2020) is a lite BERT for self-supervised learning of language representations, which uses layer-to-layer parameter sharing to reduce the number of parameters of the model, which not only speeds up the model training but also outperforms BERT on certain datasets. With our model, the pre-trained ALBERT model is fine-tuned to obtain a 512 × 768 hidden representation matrix for subsequent multi-label classification of text.

- Text-CNN (Kim, 2014) is a convolutional neural network applied to a text classification task, using multiple kernels of different sizes to extract key information in sentences, and is thus able to better capture the local relevance. In

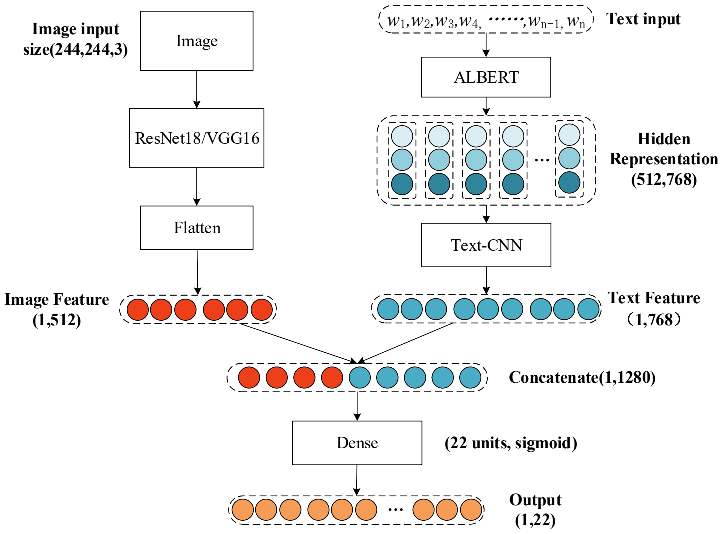

Figure 3: Architecture of parallel channel model.

this layer, we used three different sizes of one-dimensional convolution kernels, i.e., 3, 4, and 5, to extract information from the hidden representation matrix output from the ALBERT layer for the final multi-label text classification.

2.2 Subtask 2

Subtask 2 was a multi-label sequence-labeling task. We built the model by converting the problem to detect the coverage of each propagation technique separately for the input sequence, and built a multi-task sequence labeling model based on a fine-tuning of BERT.

As illustrated in Figure 3, the input sequence was first obtained using the pre-trained BERT (Devlin et al., 2019) model with a hidden representation matrix with dimensions of 512 × 768. Subsequently, 20 parallel fully connected layers were input separately for the detection of each propaganda technique coverage span (For each propagation technique, the sequence labeling task is performed separately for the input text) . For each technique, the intermediate result of each parallel channel output is a 512 × 41 matrix, and the ensemble layer represents the stacking of 20 matrices from 20 parallel channels, the dimensions of the final output were 20 × 512 × 41, which denote the propaganda technique category, maximum sentence length, and code corresponding to each technique, respectively.

2.3 Subtask 3

For subtask 3, we modeled the problem as a multi-label classification task of the meme text and image content. We used a parallel channel model of text and image channels, and then concatenated the text and image features extracted by the two parallel channels to apply multi-label meme classification. The architecture of the proposed model is shown in Figure 4.

Text Channel. In the text channel, we used the ALBERT–Text-CNN model used in subtask 1, taking the text part of the meme content as an input to obtain a 768-dimensional text feature vector as the output.

Image Channel. In the image channel, we used ResNet and VGGNet, taking the image part of the meme content as input to obtain a 512-dimensional image feature vector as the output. The ResNet model (He et al., 2016) is a deep residual learning model for image recognition, and presents the inter-layer residual jump connection and solves the deep vanishing gradient problem. VGGNet (Simonyan and Zisserman, 2015) is a deep convolutional neural network with small-sized convolutional kernels and a regular network structure, in which the size of the convolution kernels used in VGG16 in our experiment is 3 × 3, and the pooling kernels is 2 × 2. Furthermore, only the structures of the ResNet and VGGNet were used in our experiment, and the pre-training weights were not applied.

3 Experimental Results

3.1 Dataset

The organizer provided a dataset containing 687 memes for the training set, 63 memes for the development set, and 200 memes for the test set. The dataset of subtask 1 provides the ID, text of the meme, and the corresponding propaganda techniques used, and the dataset of subtask 3 also contains the corresponding meme image. The dataset of subtask 2 provides the ID, text of the meme, and the corresponding propaganda techniques used in a certain text fragment, in which the scope covered by the propaganda technology in the text is marked as "start," "end," and "text fragment," respectively.

The datasets were preprocessed using the following procedures before model training:

- In subtasks 1 and 3, we first used one-hot encoding to encode the label into a vector whose length is the total number of technology categories.
- In subtask 2, we labeled each token in the text as "I-technique" and "O-technique" based on the 20 propaganda technology terms. "I-technique" indicates that the publicity technique was used and "O-technique" indicates that it was not, e.g., O-Smears and I-Smears. For 20 different propaganda techniques there are 40 different codes, and then add another padding code, so the label code length is 41.
- In subtask 3, we normalized the meme image size to 224 × 224 × 3.

3.2 Evaluation Metrics

The official evaluation measure for all subtasks is the micro F_1-score, which is defined as follows:

$$F_1 - score = 2 * \frac{Prec * Rec}{Prec + Rec} \quad (1)$$

where *Prec* and *Rec* denote the precision and recall scores of all samples, respectively. For subtask 2, the standard micro F_1-score was slightly modified to account for partial matching between spans (Dimitrov et al., 2021). In addition, the macro F_1-score was also reported for each type of propaganda.

3.3 Implementation Details

All models used the TensorFlow2 backend, and all BERT-based models were implemented using the HuggingFace Transformers toolkit(Wolf et al., 2020). The Adam optimizer (Ba and Kingma, 2015) was used to update all trainable parameters. The loss functions in subtasks 1 and 3 were binary cross-entropy, and subtask 2 was categorical cross-entropy. The hyper-parameters in the model training process were obtained using a grid-search strategy, as shown in Table 1. Once the optimal settings of the parameters were obtained, they were used for classification on the test sets of different corpora.

Hyper-parameter	**Values**
Learning rate	5e-6
Adam epsilon	1e-8
Text-CNN dropout	0.3
Text-CNN filters	64
Classifier dropout	0.3
Batch size	16

Table 1: Hyper-parameters in our models.

3.4 Results

Table 2 presents the results of Subtask 1. We conducted experiments on several pre-trained models including BERT, RoBERTa(Liu et al., 2019), and ALBERT combined with the Text-CNN layer, and observed that the ALBERT and Text-CNN models achieved the best performance, the reason for which may be that the training datasets are small, and a serious overfitting will occur by directly fine-tuning BERT. Furthermore, the experiments show that the ALBERT model has fewer parameters and performs better on small datasets. Adding a Text-CNN layer after the BERT-based model can better extract the local relevance information of the text, which not only effectively alleviates the overfitting phenomenon it also effectively improves the model performance.

In subtask 2, the results of our proposed multi-task sequence labeling model on the dev set are F_1-score of 0.215, *Precision* of 0.378, and *Recall* of 0.151. The results on the test set are F_1-score of 0.091, *Precision* of 0.186, and *Recall* of 0.061.

Table 3 shows the results of Subtask 3. It can be observed that ResNet18 works better than VGG16 when using both ALBERT and ALBERT-Text-CNN models. The performance was improved by adding a Text-CNN layer to the text channel. Considering that the micro F_1-scores are relatively close, we selected the models with the top-three F_1-

Model	F_1-Macro	F_1-Micro
BERT	0.302	0.509
RoBERTa-Text-CNN	0.385	0.500
BERT-Text-CNN	0.414	0.560
ALBERT-Text-CNN	**0.472**	**0.625**

Table 2: Scores of different models for subtask 1 on dev set.

Model	F_1-Macro	F_1-Micro
ALBERT-VGG16	0.240	0.577
ALBERT-ResNet18	0.272	0.605
ALBERT-Text-CNN+VGG16	**0.346**	0.606
ALBERT-Text-CNN+ResNet18	0.247	**0.636**
Hard Voting	0.245	0.625

Table 3: Experimental results of different models for subtask 3 on dev set.

scores and used hard voting to generate the results for comparison.

For all three subtasks, the proposed systems achieved micro F_1-scores of 0.492, 0.091, and 0.446 on the test set, respectively. The results of all models exceeded the baseline. However, there is a considerable decrease compared to the scores of 0.625, 0.215, and 0.636 achieved on the dev set.

4 Conclusions

In this paper, we presented our system for the SemEval-2021 Task 6, the experimental results in subtasks 1 and 3 show that our proposed ALBERT-Text-CNN model and the parallel channel model achieved a good performance in the detection of persuasion techniques in texts and images.

We participated in all three subtasks and achieved the 12th, 7th, and 11th places in the test set, respectively. In a future study, to improve the generalization ability of the model, we will focus on how to deal with the problems caused by unbalanced training data.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61702443, 61966038 and 61762091.

References

Jimmy Lei Ba and Diederik P. Kingma. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR-2015)*, pages 1–15.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-2020)*, pages 1377–1414, Barcelona (online).

Jiaxu Dao, Jin Wang, and Xuejie Zhang. 2020. YNU-HPCC at SemEval-2020 task 11: LSTM network for detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-2020)*, pages 1509–1515, Barcelona (online).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2019)*, pages 4171–4186, Minneapolis, Minnesota.

Dimiter Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Task 6 at SemEval-2021: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, SemEval '21, Bangkok, Thailand.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2016)*, pages 770–778.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR-2020)*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. A survey on computational propaganda detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-2020)*, pages 4826–4832.

Andrei Paraschiv, Dumitru-Clementin Cercel, and Mihai Dascalu. 2020. UPB at SemEval-2020 task 11: Propaganda detection with domain-specific trained BERT. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-2020)*, pages 1853–1857, Barcelona (online).

Bo Peng, Jin Wang, and Xuejie Zhang. 2020. Adversarial learning of sentiment word representations for sentiment analysis. *Information Sciences*, 541:426–441.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR-2015)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS-2017)*, pages 5998–6008.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2019. Investigating dynamic routing in tree-structured LSTM for sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP-2019)*, pages 3432–3437, Hong Kong, China.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2020. Tree-structured regional cnn-lstm model for dimensional sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:581–591.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP-2020)*, pages 38–45.

Li Yuan, Jin Wang, and Xuejie Zhang. 2020. YNU-HPCC at SemEval-2020 task 8: Using a parallel-channel model for memotion analysis. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-2020)*, pages 916–921, Barcelona (online).