KSU at SemEval-2019 Task 3: Hybrid Features for Emotion Recognition in Textual Conversation

Nourah Alswaidan **Mohamed El Bachir Menai**
Department of Computer Science
College of Computer and Information Sciences
King Saud University
Saudi Arabia
nourah_swaidan@yahoo.com, menai@ksu.edu.sa

Abstract

In this paper, we present the model submitted to the SemEval-2019 Task 3 competition: contextual emotion detection in text "EmoContext". We propose a model that hybridizes automatically extracted features and human engineered features to capture the representation of a textual conversation from different perspectives. The proposed model utilizes a fast gated-recurrent-unit backed by CuDNN (CuDNNGRU), and a convolutional neural network (CNN) to automatically extract features. The human engineered features take the term frequency-inverse document frequency (TF-IDF) of semantic meaning and mood tags extracted from SinticNet. For the classification, a dense neural network (DNN) is used with a sigmoid activation function. The model achieved a micro-F1 score of 0.6717 on the test dataset.

1 Introduction

Emotion recognition in text refers to the task of automatically assigning an emotion to a text selected from a set of predefined emotion labels. The SemEval-2019 competition (Chatterjee et al., 2019b) provides a textual dialogue and asks to classify the emotion as one of the emotion labels: happy, sad, and angry or others.

Previous research shows that emotion recognition has been performed on different types of text, including fairy tales[1] (Alm et al., 2005), news headlines[2] (Strapparava and Mihalcea, 2007), blog posts[3] (Aman and Szpakowicz, 2007), and tweets[4] (Mohammad et al., 2018). Whether a text expresses a single emotion or multiple emotions, it is challenging to recognize implicit emotions, which requires natural language understanding (NLU). Recognizing emotions in textual conversation increases difficulty by adding a dialogue format. Understanding emotions in textual conversation will further boost the research on NLU.

In this paper, we present an emotion recognition model that hybridizes human engineered features and automatically extracted features. For the human engineered features, we opted for calculating the term frequency-inverse document frequency (TF-IDF) of semantic meaning and mood tags retrieved from SenticNet. For the automatically extracted features, we explored two deep neural networks, a fast gated-recurrent-unit backed by CuDNN (CuDNNGRU) and convolutional neural networks (CNN). The classification is performed by a dense neural network (DNN).

The remainder of this paper is organized as follows. Section 2 describes the task corpus. Section 3 presents the proposed emotion recognition model. Section 4 presents the experimental results, and the main conclusions and future work are presented in Section 5.

2 Corpus

The organizers of the competition split the corpus into three datasets: a training dataset with 30160 instances, a development dataset with 2755 instances, and a test dataset with 5509 instances. The corpus was in a (.txt) format and contained five columns. The first column held the ID of the instances. The second, third and fourth columns held a conversation between two individuals. The first individual started the conversation then it was the second individuals turn, then the turn returned to the first individual. The fifth column held the emotion labels of the third turn in the conversation. The emotion label was either happy, sad, angry, or others. The distribution of the emotion

[1] http://people.rc.rit.edu/ coagla/affectdata/index.html
[2] http://web.eecs.umich.edu/ mihalcea/affectivetext
[3] http://saimacs.github.io
[4] https://competitions.codalab.org/competitions/17751

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 247–250
Minneapolis, Minnesota, USA, June 6–7, 2019.

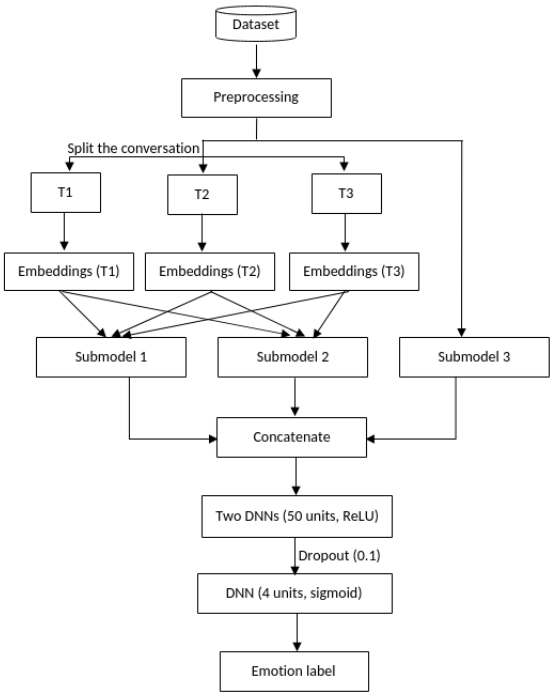

Figure 1: Diagram of the proposed model.

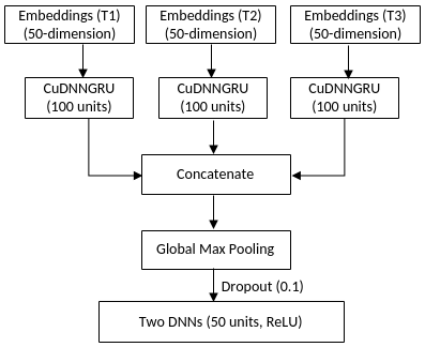

Figure 2: Diagram of submodel 1.

labels differed between the training, development and test datasets. The training data consisted of approximately 5000 instances each of happy, sad, and angry labels, and 15000 instances of the others label. The development and the test datasets had 4% each of happy, sad, and angry labels and the rest was for the label others. During the competition, the development dataset and the test dataset were released without the label column. The full development dataset was released when the final evaluation on the test dataset started. The full test dataset was released after the end of the competition.

3 Proposed Model

In this section, we present the submitted emotion recognition model. Figure 1 shows an overview of the model.

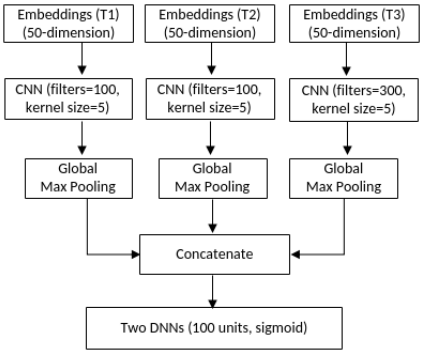

Figure 3: Diagram of submodel 2.

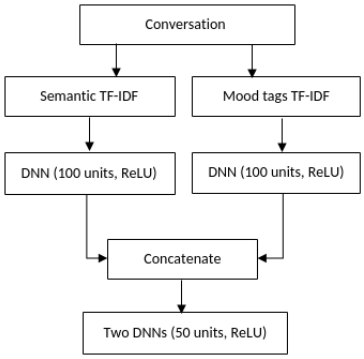

Figure 4: Diagram of submodel 3.

3.1 Preprocessing

The conversation style was informal and similar to a social media style of writing. Therefore, we utilized the ekphrasis[5] (Baziotis et al., 2017) tool. Ekphrasis was developed as part of the text processing pipeline for SemEval-2017 Task 4, sentiment analysis in Twitter. The preprocessing steps include Twitter-specific tokenization, unpack contractions, spell correction, word normalization, word annotation, word segmentation (for splitting hashtags), and replacing emoticons with suitable keywords.

We also grouped the most popular emojis into four classes, which matched the corpus emotion labels. With the use of regular expressions, we replaced the emoji with a keyword that represented the group the emoji belonged to. Then, we performed stopword removal and lemmatization with the use of the natural language toolkit[6] (NLTK).

[5]https://github.com/cbaziotis/ekphrasis
[6]https://www.nltk.org

3.2 Automatically Extracted Features

We utilized different deep neural networks, from the Keras[7] deep learning library, to enhance the representation of the text. However, we did not utilize any pretrained embeddings.

After text preprocessing, we split the text based on the conversation turns into turn 1 (T1), turn 2 (T2) and turn 3 (T3). An embedding matrix was generated for each turn of the conversation. Then, we applied BatchNormalization. These embeddings were used in two parallel submodels.

Submodel 1 in Figure 2, shows that each embedding matrix formed an input to a separate CuDNNGRU. The outputs of the three CuDNNGRUs were concatenated, and global max-pooling was performed. A dropout of value 0.1 was added to help avoid overfitting. Finally, the output was fed into two dense neural networks (DNN) with 50 units and a rectified linear unit (ReLU) activation function.

Submodel 2 in Figure 3, shows that each embedding matrix formed an input to a separate CNN with a sigmoid activation function. The number of filters of the first two CNNs was 100, but the third one had 300 filters, and the kernel size was five in all three CNNs. Next, global max-pooling was performed on the output of each CNN. Finally, the outputs were concatenated and fed into two DNNs with 100 units and a ReLU activation function.

3.3 Human Engineered Features

We took the conversation as a whole and extracted the following features:

- The TF-IDF of the Mood tags: SenticNet[8] (Cambria et al., 2018) was used to retrieve the mood tag of each word in the dataset. Then, every word was replaced by its mood tag. If a word had no mood tag, then it was deleted. Finally, the TF-IDF was calculated using the scikit-learn[9] library.

- The TF-IDF of the semantic meaning: SenticNet[8] was used to retrieve the semantic meaning of each word in the dataset. Then, the word was replaced by its semantic meaning. Finally, the TF-IDF was calculated using the scikit-learn[8] library.

[7]https://keras.io
[8]https://sentic.net
[9]https://scikit-learn.org

Item	Precision	Recall	F1
Angry	0.6345	0.8333	0.7205
Happy	0.5263	0.7746	0.6268
Sad	0.4641	0.7760	0.5808
Micro Average	0.5398	0.7962	0.6434

Table 1: Performance results on the development dataset using the automatically extracted features only.

Item	Precision	Recall	F1
Angry	0.4359	0.7933	0.5626
Happy	0.2734	0.5141	0.3570
Sad	0.4934	0.6000	0.5415
Micro Average	0.3858	0.6403	0.4815

Table 2: Performance results on the development dataset using the human engineered features only.

Item	Precision	Recall	F1
Angry	0.6531	0.8533	0.7399
Happy	0.5385	0.7887	0.6400
Sad	0.6216	0.7360	0.6740
Micro Average	0.6014	0.7962	0.6852

Table 3: Performance results on the development dataset using both automatically extracted features and human engineered features.

Item	Precision	Recall	F1
Angry	0.6456	0.7886	0.7100
Happy	0.5306	0.7324	0.6154
Sad	0.6780	0.7160	0.6965
Micro Average	0.6098	0.7476	0.6717

Table 4: Performance results on the test dataset using both automatically extracted features and human engineered features.

Submodel 3 in Figure 4, was responsible for training the human engineered features. Each of the TF-IDF features was trained with a DNN with 100 units and a ReLU activation function. Then, the outputs were concatenated and fed into two DNNs with 50 units and a ReLU activation function.

3.4 Emotion Classification

The three submodels were concatenated and fed into two DNNs with 50 units and a ReLU activation function. Then, a dropout of value 0.1 was used. Finally, a DNN with four units and a sigmoid activation function was added as an output layer for the classification of the emotions.

4 Experiments

The code was implemented in Python. We used the following libraries: NLTK[6], scikit-learn[9], and Keras[7] deep learning library run on a GPU, with the TensorFlow[10] backend.

We found the best hyper-parameters by evaluating on the development dataset. We trained with a batch size of 32, for two epochs with Adam optimization and 0.0005 as a learning rate. Tables 1 and 2 show the performance results obtained on the development dataset when only the automatically extracted features, and the human engineered features were used, respectively. They show that automatically extracted features clearly lead to the best microaverage performance results (Precision=0.5398, Recall=0.7962, F1=0.6434) in comparison to those obtained with the human engineered features only (Precision=0.3858, Recall=0.6403, F1=0.4815).

Table 3 presents the microaverage results obtained with the proposed model on the development dataset when both kinds of features were used altogether. The model achieved its best precision and F1 results (precision=0.6014, F1=0.6852) and the same recall obtained with only the automatically extracted features (Recall=0.7962). These performance results demonstrate the effectiveness of the proposed model. It scored above the baseline (Chatterjee et al., 2019a) on the test dataset. Table 4 presents the microaverage results obtained (Precision=0.6098, Recall=0.7476, F1=0.6717).

5 Conclusion

In this paper, we proposed a model to address emotion recognition in textual conversation based on using automatically extracted features and human engineered features. The usefulness of the model was demonstrated by the experimental results obtained in terms of precision, recall, and F1 measures. In the future, we plan to investigate the impact of other features on the performance of the model, including affect lexicons and pretrained embedding models.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 579–586, Stroudsburg, PA, USA. Association for Computational Linguistics.

Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *Proceedings of the 10th International Conference on Text, Speech and Dialogue*, TSD'07, pages 196–205, Berlin, Heidelberg. Springer-Verlag.

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, Vancouver, Canada. Association for Computational Linguistics.

Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. 2018. Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*, pages 1795–1802.

Ankush Chatterjee, Umang Gupta, Manoj Kumar Chinnakotla, Radhakrishnan Srikanth, Michel Galley, and Puneet Agrawal. 2019a. Understanding emotions in text using deep learning and big data. *Computers in Human Behavior*, 93:309–317.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019b. Semeval-2019 task 3: Emocontext: Contextual emotion detection in text. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019)*, Minneapolis, Minnesota.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 70–74, Stroudsburg, PA, USA. Association for Computational Linguistics.

[10]https://www.tensorflow.org