Engineering Journal issn: 0125-8281

*Article*

# EmoCNN: Encoding Emotional Expression from Text to Word Vector and Classifying Emotions—A Case Study in Thai Social Network Conversation

**Konlakorn Wongpatikaseree[1,a], Yongyos Kaewpitakkun[2,b], Sumeth Yuenyong[1,c,*], Siriwon Matsuo[3,d], and Panida Yomaboot[4,e]**

1 Department of Computer Engineering, Faculty of Engineering, Mahidol University, 25/25 Salaya, Phuttamonthon, Nakhon Pathom 73170, Thailand
2 PORDEEKUM.AI, 8/18 Pave Ramintra-wongwoen Village, Kanchanapisek Rd., O-nguen, Saimai, Bangkok 10220, Thailand
3 Sirindhorn International Institute of Technology, Thammasat University, 99 Moo 18, Km. 41 on Paholyothin Highway Khlong Luang, Pathum Thani 12120, Thailand
4 Department of Psychiatry, Faculty of Medicine Siriraj Hospital, Mahidol University, 2 Wanglang Road, Siriraj Sub-district, Bangkok-Noi District, Bangkok 10700, Thailand
E-mail: [a]konlakorn.won@mahidol.edu, [b]kaewpitakkun.yongyos@gmail.com, [c,*]sumeth.yue@mahidol.edu (Corresponding author), [d]siriwon.matsuo@gmail.com, [e]panida.yomaboot@gmail.com

**Abstract.** We present EmoCNN, a collection of specially-trained word embedding layer and convolutional neural network model for the classification of conversational texts into 4 types of emotion. This model is part of a chatbot for depression evaluation. The difficulty in classifying emotion from conversational text is that most word embeddings are trained with emotionally-neutral corpus such as Wikipedia or news articles, where emotional words do not appear very often or at all, and the language style is formal writing. We trained a new word embedding based on the word2vec architecture in an unsupervised manner and then fine-tuned it on soft-labelled data. The data was obtained from mining Twitter using emotion keywords. We show that this emotion word embedding can differentiate between words which have the same polarity and words which have opposite polarity, as well as find similar words with the same polarity, while the standard word embedding cannot. We then used this new embedding as the first layer of EmoCNN that classifies conversational text into the 4 emotions. EmoCNN achieved macro-averaged f1-score of 0.76 over the test set. We compared EmoCNN against three different models: a shallow fully-connected neural network, fine-tuning RoBERTa, and ULMFit. These got the best macro-averaged f1-score of 0.5556, 0.6402 and 0.7386 respectively.

**Keywords:** Emotion classification, sentiment analysis, word embedding.

**ENGINEERING JOURNAL** Volume 25 Issue 7
Received 16 December 2020
Accepted 1 July 2021
Published 31 July 2021
Online at https://engj.org/
DOI:10.4186/ej.2021.25.7.73

DOI:10.4186/ej.2021.25.7.73

## 1. Introduction

Depression is a global health issue. The World Health Organization states in a report [1] that depression affects around 300 million people worldwide. For Thailand, the Department of Mental Health reports that out of the 53 million people aged 15 and above, around 1.44 million are suffering from depression [2]. Depression is characterized by mood alteration, loss of appetite, constant feeling of sadness/loneliness, sleep disorder, and in severe cases suicidal thought/attempt. It can also lead to social and economic problems because sufferers are likely to lose interest in doing their daily activities, impacting their jobs, and have the desire to escape or disappear, affecting their families [3].

Depression is not an individual problem. A sufferer's family also shares the burden, both socially and economically. It is estimated that by 2030, depression could cost the world economy as much as $16 trillion[1]. More importantly, depression is a major cause of suicide [1, 4], even though it can be prevented and treated. It is important to be able to detect depression in the early stage so that it can be treated or at least managed before it becomes a serious condition, minimizing the impact that it would have on a sufferers' lives and families. A study [5] found that detecting depression early helped decrease prevalence and improve treatment response. This was true for people of all age groups and was especially important for those with other pre-existing medical conditions [6].

The tools available to health care providers for depression screening are generally in the form of self-rating questionnaire. For Thai, one of the most widely-used screening is the Thai Mental Health Questionnaire (TMHQ) [7]. It is DSM-IV[2] based and is 70-questions long. The length of the questionnaire creates a problem of getting complete responses, as a common symptom of depression is lack of motivation and low energy level. Recently, the psychology literature have had many studies about detecting depression through online behavior such as [8, 9]. This approach offers important advantages over traditional questionnaire in that a test can be administered to a large number of people efficiently and participants are less likely to give responses with social-desirability bias [10] since the test is anonymous and they are not directly communicating with a person or even have to leave their homes.

A very important part of detecting depression through online behavior is the ability to correctly classify emotions behind texts (posts, tweets, chat messages, etc.), as negative emotions play a key role leading to depression. People who suffer from prolonged negative emotions are likely to fall into depression [3]. However, detecting emotion from just text without actually having a conversation with the person is a challenging task. Part of the difficulty is that normally, word embeddings such as word2vec [11] are not trained on texts which have emotional connotations, but rather "dry" text such as Wikipedia – a popular choice for standard ready-to-use word embeddings that can be downloaded in many non-English languages, including Thai.

In this work we present *EmoCNN* which consists of two parts: the first is a word embedding based on the word2vec architecture which is specially trained to be sensitive to emotional words. The second part is a classification model that classifies texts into 4 different types of emotions: positive activation – positive and "active" (e.g. excited, joy), positive deactivation – positive and "passive" (e.g. peaceful, relieved), negative activation (e.g. stressed, angry) and negative deactivation (e.g. sad, bored). These 4 types form the basis for the emotion wheel [12] and the circumplex model of emotion [13] which in turn is an important part of depression evaluations. The classification model is a convolutional neural network [14] that gets its input from the output of the embedding layer. It consists of several convolutional layers and 2 fully-connected layers working as classification head. The trained model is currently deployed as part of a chatbot system also developed by the authors called "Chujai" – a chatbot for assesing the emotional well-being of a person through natural conversation as opposed to formal questionnaires.

### 1.1. Labeling Emotion Behind Text

The emotion groups that we considered are base on the circumplex model of emotion [13] which organize emotions long two axes: valance and arousal. Valance is basically positive/negative and arousal is how "active" the emotion is. So given a text, first it is determined as positive/negative by looking at keywords such as happy, excited, bored, tired and by considering the meaning of the text as a whole. For example, a rant on social media about something he/she doesn't like is negative, whereas a text talking about how the person just received a gift is positive. This gives the placement on the valance axis. Note that we do not consider the strength of the emotion, so positive/negative is binary classification. For arousal, "active" verbs are considered. These words signal activated emotions of either positive or negative. For example, "hit" and "cry" are active verbs for negative emotion whereas "smile" and "dance" are active verbs for positive emotion. These verbs have visible actions that other people could see. On the other hand, text with "passive" verbs such as "bored" and "disappointed" which lack visible actions are generally deactivated emotions. Moreover, every label is cross-checked with at least one other labeler.

## 2. Related Works

A problem that is similar to emotion classification is sentiment analysis, where the objective is to classify a piece of text as positive/negative. Use cases include

[1] https://www.reuters.com/article/us-health-mental-global/mental-health-crisis-could-cost-the-world-16-trillion-by-2030-idUSKCN1MJ2QN

[2] https://en.wikipedia.org/wiki/Diagnostic_and_Statistical_Manual_of_Mental_Disorders

movie or restaurant reviews, customer chat logs, comments on social media and more recently chatbots. Both sentiment analysis and emotion classification fall under the bigger umbrella of text classification, which is one of the major problem types in Natural Language Processing (NLP). Traditionally, text classification was performed using statistical features such as TF-IDF [15], or problem-specific rules and keywords-based approach such as in [16]. With the resurgent of neural networks and the rise of deep learning in recent years, many researchers applied deep learning to the task of text identification. One of the first was [17] where the author applied convolutional neural network to classify texts. Later [18] used both CNN and recurrent neural network for sentiment analysis on Twitter data. The main advantage of the neural network approach for text classification is that there is no need for manual feature engineering and one model is applicable to a wide range of problems – only the training data has to change. As such, it is a commonly-used approach nowadays.

Considering text classification in Thai in particular, similar to this work is [19] where the author used a combination of term weighting and Support Vector Machine (SVM) to classify Thai sentences into 6 different emotions. The data used were posts from popular web boards and social media sites. Later [20] considered the same group of emotions but specifically from Youtube comments. Both of these works are similar in the sense that they both used manual feature extraction followed by shallow classifier such as SVM or Naive Bayes, which require feature engineering. More recently, [21] used deep learning for classification of Thai tweets as positive or negative. The architecture used was similar to our EmoCNN here, but the number of classes was less and the authors did not consider the impact of the embedding layer. EmoCNN is unique in that we gave the embedding layer careful consideration rather than just use it off-the-shelf, the emotions considered are backed by psychology literature, our training data was labelled by a professional psychologist and we compared our results with different techniques which are quite recent.

## 3. EmoCNN

The high-level block diagram of EmoCNN is shown in Fig. 1. The doll represents our Chujai chatbot. The top block represents the training data for the emotion word embedding. Once trained, the new embedding is used as the first layer of the classifier model. The classifier itself is basically a CNN consisting of 1D convolutions and pooling layers. The target cases of the paper are firstly, to train the embedding layer so that it is able to better capture the emotion behind single words, and secondly to train a classifier together with the new embedding to maximize emotion classification performance. The details of how the Chujai chatbot reply to the user is beyond the scope of the paper, but can be described briefly here: we have a database of text-responses pairs (e.g. "how are you" - "I am fine") for each of the four emotions, as a new text from the user arrive, it is classified into one of the four emotions, then we search for the most similar text (within that emotion only) and return its response pair to the user. The detail for the model architecture will be explained below.

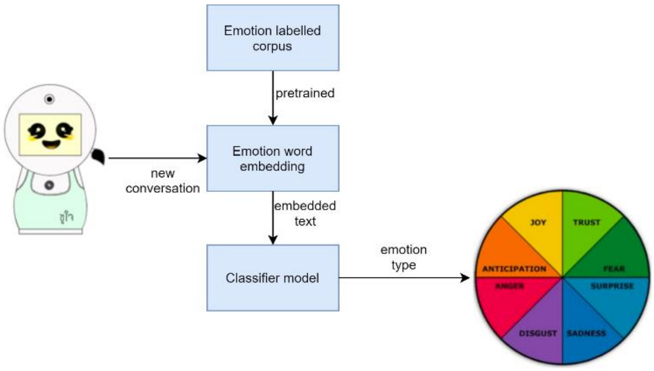


Fig. 1. The block diagram of EmoCNN.

As previously stated, the main limitation of using a standard pre-trained word2vec (or any other word embedding) directly is that they are typically trained on emotional-neutral Wiki or news articles text, in which emotional words are rare or not found. Therefore the embedding layer is not able to capture the meaning of these words well like in the famous king-man+woman=queen example. Put another way, standard word embeddings are not sensitive to emotion words. This can negatively impact the performance of a downsteam classifier. Figure 2 illustrates this issue. The left-hand side shows an embedding that cannot distinguish between the opposite polarity of words, because words such as "love" or "hate" rarely appears in the data used to train them. The right-hand side shows an embedding that can distinguish between positive emotions and negative emotions. The embedding layer of EmoCNN is closer to the right-hand side.

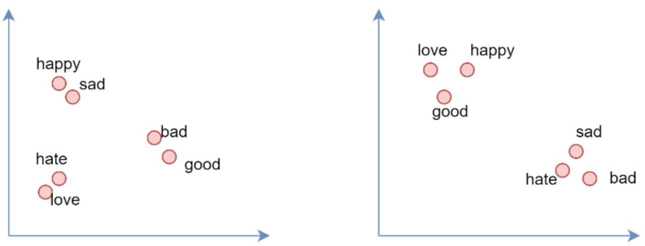


Fig. 2. (Left) an embedding which cannot separate emotionally opposite words. (Right) an embedding which is sensitive to the emotional meaning of words.

### 3.1. The Training Data for Emotion Word Embedding

We prepared a list of keywords for Twitter search by listing out words in different emotion categories such as: happy, sad, delighted, longing, jealous, etc. In total we had over 200 keywords. We mined Twitter using each keyword, then tokenized and cleaned each tweet that we obtained using standard NLP techniques for preprocessing [22] such as removing stop words, normalization, stemming, and removing onomatopoeia. This process was done mainly by regular expression and the PyThaiNLP library[3]. For tokenization, we used the cutkum Python library[4]. In total, we collected and processed about 5 million tweets.

### 3.2. Training the Emotion Word Embedding

After obtaining the data we trained word2vec from scratch using the standard configuration as in [11], which is the continuous bag of word model (CBOW) – where the training task is to predict the middle word, given the context consisting of the previous two words and the next two words. The vocabulary size was 100,000 and embedding size was 100. The embedding layer trained in this manner is basically the same as the standard word2vec, except for the training data. We called this embedding 1. We tested whether this word embedding can separate between emotionally-positive/negative words, compared to the pre-trained embedding of PyThaiNLP (called "Thai2vec"). The comparison was done by embedding each word and then calculating the cosine similarity between pairs of them. Some examples of the results for opposite-polarity pairs are shown in Tables 1 and 2. It can be seen that the standard word embedding just treats all emotional words as the same, assigning very high similarity between all pairs. This is likely because these words don't appear very often in Wikipedia or news articles that they just get grouped together as unknown words. For embedding 1, it can be seen that even though the cosine similarity is lower, indicating that the model no longer treats all these words as the same, the difference between opposite pairs and similar pairs were not sufficient to be able to distinguish between emotionally positive/negative words based on the cosine similarity values.

It was necessary to fine-tune embedding 1 in a supervised manner in order to improve the result. We labelled the 5 million tweets that we obtained as positive/negative. This was impractical to do manually, so we used emoticon as soft labels. Tweets that have positive emoticon such as smiling face or similar were consider positive and tweets that has crying face or similar were considered negative. Tweets that do not have any emoticon were excluded. After labelling was completed, all emoticons were removed before training. The number of tweets left after this process were 367,709 for negative emotions tweets and 156,856 for positive emotions tweets. A CNN model (note that this is not the final EmoCNN, it was only temporary for fine-tuning the embedding layer) was built using embedding 1 as the initial weight for the embedding layer. The rest of the model had 3 layers of 1D convolution, followed by some dropout [23] and fully connected layers. The output layer was for binary classification (positive/negative). The exact model architecture is shown in Fig. 3.

Table 1. Cosine similarity between pairs of opposite-meaning words. We abbreviated "embedding" as emb. to save space.

| Words (opposite) | Thai2vec | emb. 1 | emb.2 |
|---|---|---|---|
| excited vs. bored | 0.9960 | 0.3549 | -0.1632 |
| happy vs. sad | 0.9881 | 0.3603 | -0.1155 |
| satisfied vs. stressed | 0.9952 | 0.2881 | -0.2579 |
| relaxed vs. angry | 0.9919 | 0.1544 | -0.0712 |

Table 2. Cosine similarity between pairs of similar-meaning words. We abbreviated "embedding" as emb. to save space.

| Words (similar) | Thai2vec | emb. 1 | emb.2 |
|---|---|---|---|
| excited vs. happy | 0.9974 | 0.3079 | 0.4748 |
| satisfied vs. relaxed | 0.9943 | 0.1085 | 0.3247 |
| bored vs. sad | 0.9944 | 0.4231 | 0.6763 |
| stressed vs. angry | 0.9940 | 0.3684 | 0.5536 |

The fine-tuned embedding layer is the "emb. 2" columns in Tables 1 and 2. It can be seen that the cosine similarities were sufficiently different between similar-pairs and opposite-pairs. This indicates that embedding 2 was capable of distinguishing between emotionally positive/negative words.

[3] https://github.com/PyThaiNLP/pythainlp
[4] https://github.com/pucktada/cutkum

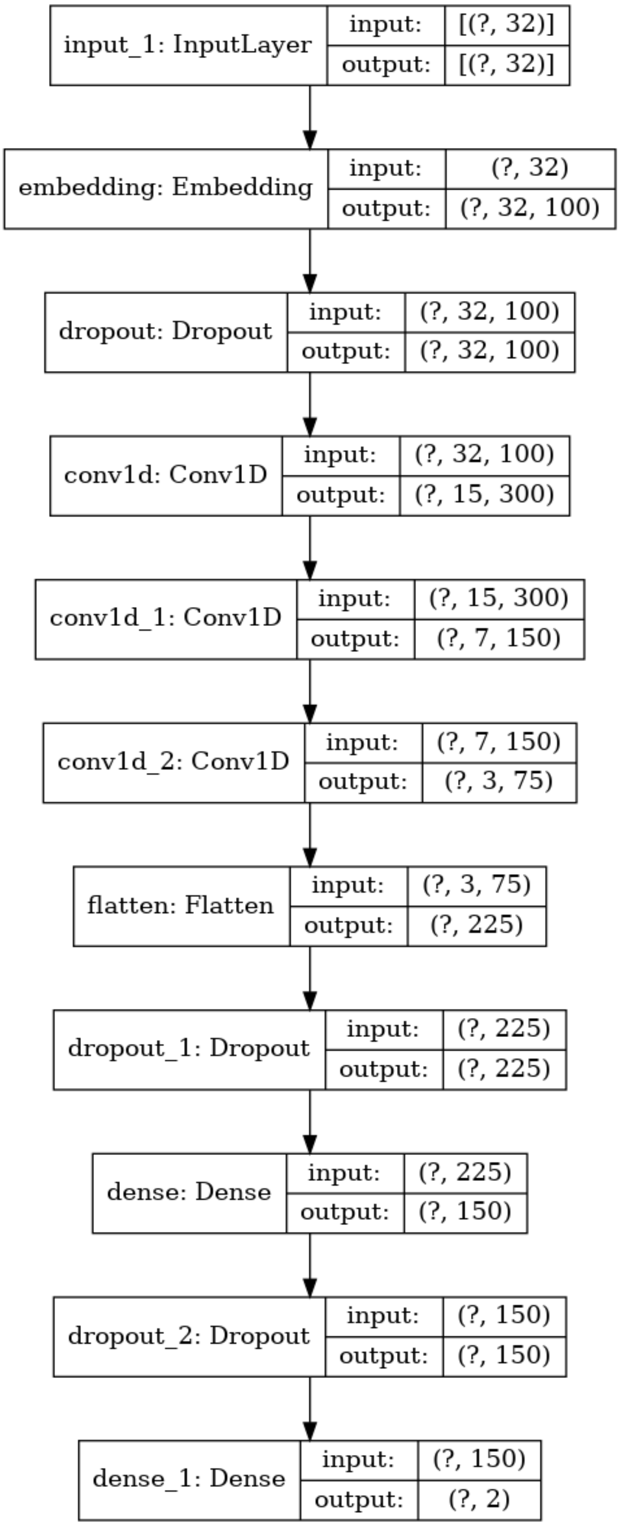


Fig. 3. The model architecture for fine-tuning the emotion embedding layer. Note the second layer is loaded from the pre-trained embedding 1. The "?" means that the size of this dimension is determined by the batch size set at training time.

### 3.3. Finding Similar Words Using the Emotion Word Embedding

In this section we present some results of using the fine-tuned embedding 2 (which is the actual proposed emotion word embedding) to find words which have similar meaning (positive/negative) to a query word based on cosine similarity value. We searched for the top 20 most similar words to a query word using both the standard word embedding and our embedding 2 and found that for the standard word embedding, some results are clearly not similar. For example, when the query word was "sad", the standard word embedding got words such as "excited", "happy", "cute", and "like". On the other hand, our embedding 2 got words with negative connotation for all 20 words. When the query word was "bored" standard word embedding got words such as "trust", "like", and "satisfied". The embedding 2 got words such as a synonym of bored itself, along with words like "hate", "sad", "uncomfortable", and "annoyed". For the query tested, embedding 2 was always able to find 20 words with the same polarity and similar connotation as the query word. We show one example output for a query word "satisfied" in Fig. 4. On the left hand side are the words obtained by standard word embedding and on the right are the words obtained by our embedding 2. The words that are underlined in red are words that have opposite polarity to the query word. Also note that many words (on both sides) are slang/colloquial words which are not found in Wiki or news articles. This accentuates the need to pre-train and fine-tune embedding layers on domain-specific texts for best results.

### 3.4. Emotion Classification

The final part of our work here is the emotion classification models. To build the training data for this task, we sampled sentences from the training set that we used to fine-tune embedding 2, and one of the authors who is a psychologist labelled them into the four classes: positive activation (e.g. excited, joy), positive deactivation (e.g. peaceful, relieved), negative activation (e.g. stressed, angry) and negative deactivation (e.g. sad, bored). In total we had about 4,000 training examples for the classification model.

The architecture of the classification model was similar to [24]. It was a CNN where the first layer's weights were loaded from embedding 2 in the previous section. This was followed by 3 separate branches of 1D convolution and global max pooling. Each branch had different kernel sizes for the 1D convolution layer: 2, 3, and 4 respectively. This corresponds to considering words as bigram, trigram and 4-gram units. The global max pooling was there to ensure that the output of each branch had compatible shapes to be concatenated together. After merging the 3 branches together the model had some fully connected and dropout layers that perform the actual classification. The detailed architecture is shown in Fig. 5.


Fig. 4. The top 20 words that are most similar to the query word "satisfied". (Left) the result from standard word embedding. (Right) the result from embedding 2. The red underlined words have negative polarity which is opposite from "satisfied".

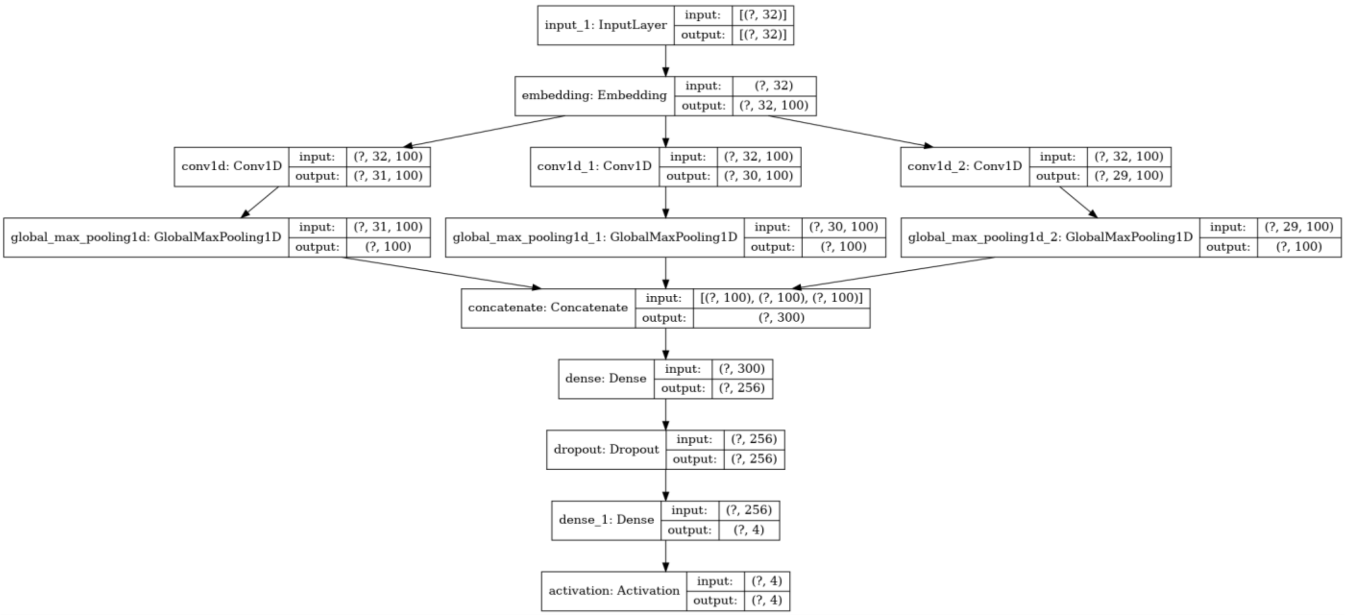


Fig. 5. The architecture of the emotion classification model.

For training/testing, we randomly split the data into 80% train and 20% test. We trained the model for 100 epochs using the default parameters for the Adam optimizer in Keras. The result for both 2-class (positive/negative) setting and the full 4-class setting are shown in Tables 3 and 4 respectively. For the 2-class case, the macro-average f1-score was 0.89. For the 4-class case it was 0.76.

Table 3. Result for 2-class (positive/negative) emotion classification.

| class | precision | recall | f1 | support |
|---|---|---|---|---|
| neg | 0.87 | 0.91 | 0.89 | 393 |
| pos | 0.91 | 0.86 | 0.89 | 409 |
| macro avg | 0.89 | 0.89 | 0.89 | 802 |

DOI:10.4186/ej.2021.25.7.73

Table 4. Result for 4-class emotion classification.

| class | precision | recall | f1 | support |
|---|---|---|---|---|
| neg act. | 0.69 | 0.69 | 0.69 | 207 |
| neg deact. | 0.68 | 0.76 | 0.72 | 186 |
| pos act. | 0.86 | 0.77 | 0.81 | 210 |
| pos deact. | 0.82 | 0.82 | 0.82 | 199 |
| macro avg | 0.76 | 0.76 | 0.76 | 802 |

As a test example, we crafted new sentences "I saw in my friend's Instagram story that she is happy. I am happy that she's happy" and "I saw in my friend's Instagram story that she is happy, I should be happy that she's happy ... but I'm not.". The first sentence is positive-deactivation (happy, relieved) while the second is negative-deactivation (sad, bored) Both were correctly classified by EmoCNN. A screenshot of the output is shown in Fig. 6. What's remarkable about this result is that the second sentence contains many "happy" but it's still correctly classified as negative-deactivation.

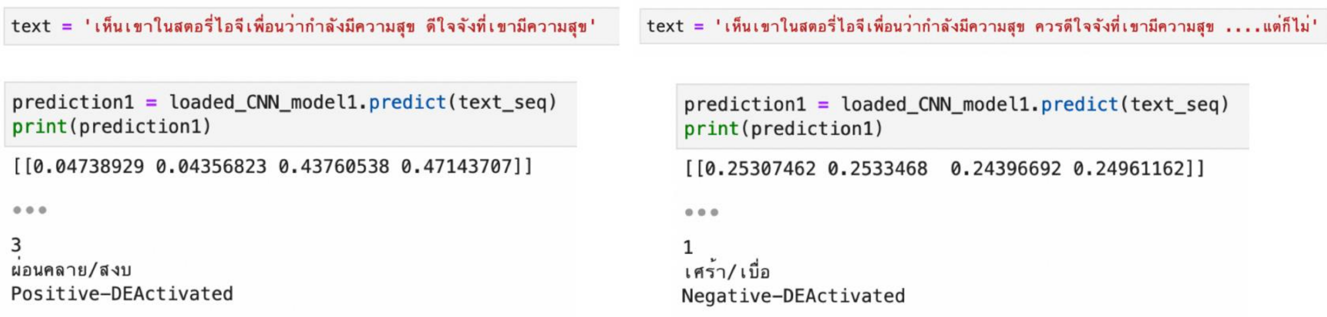


Fig. 6. (Left) result of the first sentence. (Right) result of the second sentence.

In order to put the performance of EmoCNN in context, we trained three different models for comparison. The first was a multi-layered perceptron (MLP) architecture consisting of 3 layers with sizes 256, 65, and 4. The training sentences was pre-embedded into vector space using standard embedding layer to embed each word, and then adding the vector of each word together to get the vector representation of the whole sentence. We trained this model for the same number of epochs using the same training hyperparameters as EmoCNN. The best f1-score it achieved was 0.5556. The second model was RoBERTa [25] which is an improvement upon the famous BERT [26]. We used the base-sized version (12 transformer layers with 12 attention heads each). We pretrained RoBERTa on the same set of tweets that we used to pretrain our embedding 1 and then fine-tune it on the same data as EmoCNN. For this model we got the highest f1-score of 0.6402. Although transformer architectures are now state-of-the-art in NLP, here we have too little data to train it properly. Our entire pre-train tweet data consists of only 80 MB, while the original BERT was train on 13 GB of data and RoBERTa on 160 GB. Not being able to pre-train well due to limited data likely impac the performance of downstream fine-tuning. Moreover, the size of BERT/RoBERTa base is about 100M parameters, while EmoCNN has less than 10M. Smaller models tend to perform better when the data is limited [27].

The final model for comparison was the ULMFit model [28]. ULMFit is both a model architecture and a training paradigm. The architecture is AWD-LSTM [29] while the training paradigm mainly concerns the manipulation of the learning rate layer-wise, where each layer has different learning rates, and epoch-wise, where the learning rates increase and then decrease with the epoch, also known as triangular learning rate. ULMFit is considered by many to be the "go to" strong baseline for sentence classification because the model is relatively small, offers strong performance, and fast easy-to-use implementation is available from the fast.ai[5] library [30]. ULMFit achieved best f1-score of 0.7386 on the same data as EmoCNN. Table 5 summarizes the performance of the three competing models as well as EmoCNN.

Table 5. The macro-averaged f1-score of the three competing models and EmoCNN.

| | architecture | f1-score |
|---|---|---|
| model 1 | MLP | 0.5556 |
| model 2 | RoBERTa | 0.6402 |
| model 3 | ULMFit | 0.7386 |
| **ours** | **EmoCNN** | **0.7600** |

## 4. Further Work

An interesting research direction is to experiment with contextualized word embeddings such as ELMO [31] where other words in the sentence is incorporated into the embedding of each word. Another avenue that might improve the performance further is hierarchical models (classify positive/negative first, followed by actual class) and/or using ensemble techniques.

[5] https://www.fast.ai/

Dimension reduction of the features (the activation in the model just before the first fully-connected layer) using techniques such as the one proposed recently in [32] could also be considered.

## 5. Conclusion

In this work we presented EmoCNN: embedding layer plus CNN for classifying conversational texts into 4 emotions. Standard word embedding cannot capture the meaning of emotional words very well due to their training data. Our emotion word embedding overcome this by training and then fine-tuning on text mined from Twitter using emotional keyword search. The word embedding was able to distinguish between words with opposite polarity, as well as find similar words with the same polarity. We then used this emotion word embedding as the first layer in a CNN model for classifying emotions from text. The model consists of 3 convolutional branches with different kernel sizes to look at text at different-sized units. The output of the branches was concatenated together for final classification. We compared EmoCNN against 3 different models. The closest was ULMFit with f1-score of 0.7386 vs. 0.76 for EmoCNN. We have already integrated EmoCNN into the Chujai chatbot for depression screening.

## Acknowledgement

This study is a part of a larger research which is fully granted by the Office of The National Broadcasting and Telecommunications Commission grant number B2-114/1-61. We would also like to express our thanks to the ethical committees of Mahidol University for granting an ethical approval number MU-CIRB 2017/142.1108.

## References

[1] World Health Organization, "Depression and other common mental disorders: global health estimates," World Health Organization, Tech. Rep., 2017.

[2] Thailand Department of Mental Health, "Annual report fiscal year 2019," Department of Mental Health, Tech. Rep., 2020.

[3] A. T. Beck and B. A. Alford, *Depression: Causes and Treatment*. University of Pennsylvania Press, 2009.

[4] M. Marcus, M. T. Yasamy, M. van Ommeren, and D. Chisholm, "Depression: A global public health concern," WHO Department of Mental Health and Substance Abuse, Tech. Rep., 2012.

[5] E. Garland and K. Solomons, "Early detection of depression in young and elderly people," *British Columbia Medical Journal,* vol. 44, no. 9, pp. 469-472, 2002.

[6] J. W. Feightner and G. Worrall, "Early detection of depression by primary care physicians," *CMAJ: Canadian Medical Association Journal*, vol. 142, no. 11, p. 1215, 1990.

[7] S. Phattharayuttawat, T. Ngamthipwattana, and K. Sukhatungkha, "The norm profile for 'the Thai mental health questionnaire'," *Journal of the Psychiatric Association of Thailand*, vol. 15, no. 20, p. 212, 1999.

[8] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz, "Predicting depression via social media," in *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[9] X. Wang, C. Zhang, Y. Ji, L. Sun, L. Wu, and Z. Bao, "A depression detection model based on sentiment analysis in micro-blog social network," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2013, pp. 201–213.

[10] H. J. Arnold and D. C. Feldman, "Social desirability response bias in self-report choice situations," in *Academy of Management Journal*, vol. 24, no. 2, pp. 377–385, 1981.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[12] R. Plutchik, "A general psychoevolutionary theory of emotion," in *Theories of Emotion*. Elsevier, 1980, pp. 3–33.

[13] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, p. 1161, 1980.

[14] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, p. 1995, 1995.

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[16] R. Batool, A. M. Khattak, J. Maqbool, and S. Lee, "Precise tweet classification and sentiment analysis," in *Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on*, IEEE, 2013, pp. 461–466

[17] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.

[18] M. Cliche, "BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 573–580. [Online]. Available: https://www.aclweb.org/anthology/S17-2094.

[19] N. Chirawichitchai, "Emotion classification of Thai text based using term weighting and machine learning techniques," in *2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, 2014, pp. 91–96.

[20] P. Sarakit, T. Theeramunkong, C. Haruechaiyasak, and M. Okumura, "Classifying emotion in Thai YouTube comments," in *2015 6th International Conference of Information and Communication Technology*

*for Embedded Systems (IC-ICTES)*, IEEE, 2015, pp. 1–5.

[21] P. Vateekul and T. Koomsubha, "A study of sentiment analysis using deep learning techniques on Thai twitter data," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, 2016, pp. 1–6.

[22] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., 2009

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, *arXiv:1510.03820*.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[27] A. Ezen-Can, "A comparison of LSTM and BERT for small corpus," 2020, *arXiv:2009.05451*.

[28] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018, *arXiv:1801.06146*.

[29] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models," 2017, *arXiv:1708.02182*.

[30] J. Howard and S. Gugger, "Fastai: A layered API for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020.

[31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*.

[32] A. Dixit, A. Mani, and R. Bansal, "Feature selection for text and image data using differential evolution with SVM and naïve Bayes classifiers," *Engineering Journal*, vol. 24, no. 5, pp. 161–172, 2020.

**Konlakorn Wongpatikaseree** received his B.Sc. from Thammasart University in 2008. During 2008–2010, he received a scholarship to pursue a Master degree from the Thailand Advanced Institute of Science and Technology and Tokyo Institute of Technology. He received his Ph.D. in information science from Japan Advance Institute of Science and Technology. Now he is a lecturer at Department of Computer Engineering, Faculty of Engineering, Mahidol University, Thailand. His research interests are in intelligent systems, embedded systems, smart homes, activity recognition, and healthcare systems. He is a member of IEICE.

**Yongyos Kaewpitakkun** received PdD degree from Japan Advanced Institute of Science and Technology, Japan in 2016. His research interests lie in the field of natural language processing and artificial intelligent.

DOI:10.4186/ej.2021.25.7.73

**Sumeth Yuenyong** received his B.Eng. in Electrical Engineering from Sirindhorn International Institute of Technology, Thammasat University in 2005, M.Eng. and D.Eng. in Communication and Integrated Systems from Tokyo Institute of Technology in 2011 and 2014 respectively. He received MEXT scholarship in 2010. He is now a lecturer at the Department of Computer Engineering, Faculty of Engineering, Mahidol University, Thailand. His research interests are signal processing, artificial intelligence and natural language processing.

**Siriwon Matsuo** is currently a Data Scientist at LINE Company (Thailand). She received her bachelor's degree and master's degree in Computer Engineering from Kasetsart University, Thailand, and dual doctoral degrees in Knowledge Science from Japan Advanced Institute of Science and Technology, Japan, and Engineering and Technology from Sirindhorn International Institute of Technology, Thammasat University, Thailand. Her research interests are natural language processing, data mining, text mining, machine learning, and recommender system.

**Panida Yomaboot** is a licensed clinical psychologist at the Department of Psychiatry, Faculty of Medicine Siriraj Hospital, Mahidol University. Her research interests are in clinical psychology, depression, personality, psychometric property, music psychological and the application of AI in psychology. She has a strong background in psychology as she holds B.Sc. in Psychology with a second class honour, M.Sc. in Clinical Psychology and she is working toward the final step of her PhD in psychology at Goldsmiths, University of London.