Sentence Embedding Based Emotion Recognition from Text Data

Manabu Ito
University of Aizu
Aizuwakamatsu, Fukushima, Japan
m5261119@u-aizu.ac.jp

Konstantin Markov
University of Aizu
Aizuwakamatsu, Fukushima, Japan
markov@u-aizu.ac.jp

ABSTRACT

Automatic emotion recognition from text is an important task in the field of natural language processing (NLP) with applications in data mining, e-learning, information filtering systems, human-computer interaction, and internet services. Recent advances in machine learning and deep neural networks have boosted the NLP systems' performance tremendously. The availability of large pre-trained language models has simplified the feature extraction and facilitated the building of systems from small amounts of data by transfer learning. In this study, we investigate and compare various methods of sentence embedding including simple embedding matrix as well as sophisticated models such as BERT. The recognition back-end consists of standard SVR or Feed-Forward DNN regressors. In our experiments, we used the EmoBank corpus where each sentence is labeled with Valence-Arousal scores. The results clearly show the benefits of the transfer learning and fine-tuning of pre-trained models with respect to classical model training from scratch.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**.

KEYWORDS

Emotion recognition, Sentence embedding , Neural text processing

ACM Reference Format:

Manabu Ito and Konstantin Markov. 2022. Sentence Embedding Based Emotion Recognition from Text Data. In *International Conference on Research in Adaptive and Convergent Systems (RACS '22), October 3–6, 2022, , .* ACM, New York, NY, USA, Article 4, 5 pages. https://doi.org/10.1145/3538641.3561488

1 INTRODUCTION

With the recent popularity of social media, data from sources such as audio, text, images, or video became available for conducting studies in various areas of human life and behavior. Social media posts, micro-blogs, news articles, etc., are useful resources for mining and collecting text data for analysis and predicting people's personality traits and emotions. Analyzing emotions is helpful in many different domains. One such domain is human-computer interaction. With the help of emotion recognition, computers can make better decisions to help users or provide better social services.

RACS '22, October 3–6, 2022, Virtual Event, Japan

ACM ISBN 978-1-4503-9398-0/22/10…$15.00
https://doi.org/10.1145/3538641.3561488

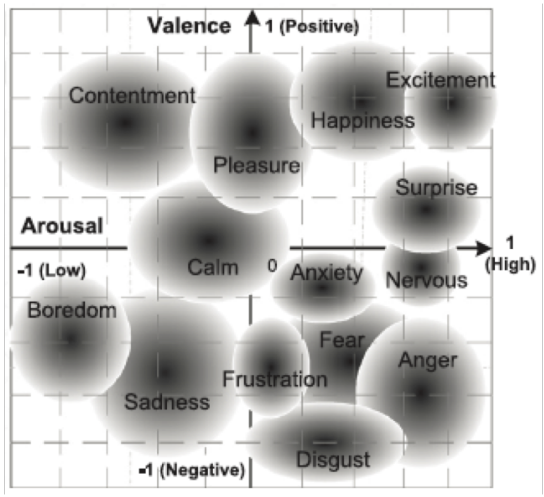

Figure 1: Two dimensional (Valence-Arousal) affective space of emotions [4]. Different regions correspond to different categorical emotions.

With the increase in the popularity of robotic research, emotion recognition will also help make human-robot interaction more natural

Affective computing [5, 7] is science under which methods are being developed that can process, identify and understand human emotions. For the last three decades, a large number of methods are continuously being devised to facilitate emotion analysis; from manual methods such as questionnaires elaborated by psychologists to methods involving computers and AI models [16].

Most of the automatic systems for text emotion recognition are based on emotion representation which can be either categorical or dimensional [2, 15, 18]. Categorical approaches involve finding emotional descriptors, usually adjectives, which can be arranged into groups. Given the perceptual nature of human emotion, it is difficult to come up with an intuitive and coherent set of adjectives and their specific grouping. To alleviate the challenge of ensuring consistent interpretation of mood categories, some studies propose to describe emotion using continuous multi-dimensional metrics defined in low-dimensional spaces. Most widely accepted is Russell's two-dimensional Valence-Arousal (VA) space [14] where emotions are represented by points in the VA plane. Figure 1 shows the space where some regions are associated with distinct mood categories.

Previous research on affective computing has merely utilized methods from traditional machine learning, while recent advances from the field of deep learning have just made an inroad in the latest studies. Lexicon-based approaches utilize pre-defined lists of terms that are categorized according to different affect dimensions [11].

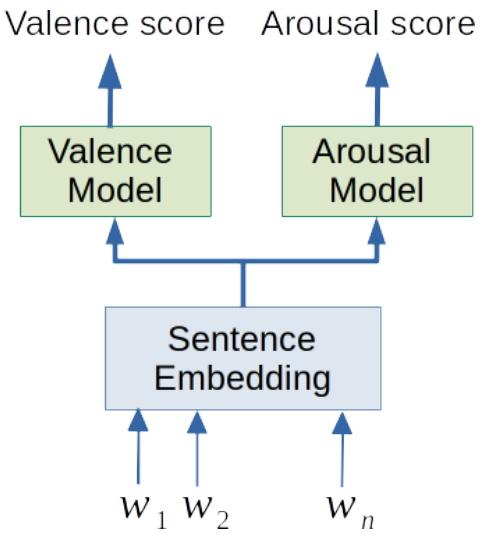

Figure 2: Block diagram of the emotion recognition system.

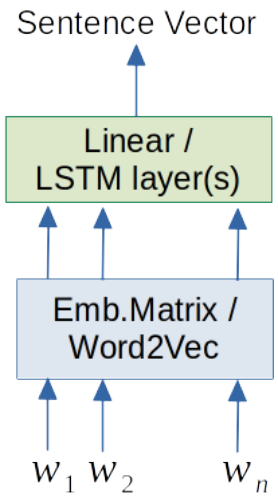

Figure 3: Sentence embedding using word Embedding Matrix or Word2Vec and Linear or LSTM neural network layers.

On the one hand, these lexicons are often compiled manually, a fact that can later be exploited for keyword matching. For instance, the Harvard IV dictionary (inside the General Inquirer software) and LIWC provide such lists with classification by domain experts [17].

Recent approaches based on deep neural networks usually apply LSTM models to capture the sentence context [8, 9]. However, transfer learning and large language models such as BERT have been used lately with great success [1]. In this work, we study and compare the performance of word-level and sentence-level embedding methods using model architectures including LSTM and several BERT variants.

2 SYSTEM DESCRIPTION

Our system takes as an input one sentence, i.e. a sequence of words $w_1, w_2, ..., w_n$ and predicts the Valence and Arousal scores for that sentence. The block diagram of the system is shown in Fig. 2. It consists of two main parts: front-end sentence encoder and back-end regression models, one for the Valence and one for the Arousal prediction. The sentence encoder transforms the input words into a single vector which is then passed to the back-end models.

The focus of this study is the performance of several different sentence embedding methods which have recently gained popularity among researchers. They can be broadly divided into two categories: 1) word embedding-based methods, and 2) direct sentence embedding methods. The approach in the first category is to obtain embedding vectors for each word in a given sentence and then combine those vectors into one. In contrast, in the second category of methods, the whole sentence is directly transformed into a vector using some language model. To provide a fair and unbiased comparison of all those methods, as a back-end we use simple SVR and Feed-Forward DNN models.

2.1 Word embedding based methods

Transforming words into vectors is the basis of all current natural language processing methods. This procedure embeds each word into a high dimensional space and the way this space is constructed depends on the particular transformation method. One of the first and most popular approaches is the Word2Vec [10] where the model is trained to predict a word from its neighbors using the Continuous Bag-of-Words (CBoW) technique. Word vectors can also be derived from the Skip-gram model where the surrounding words are predicted from a given word. These models are usually trained from a large text corpus, but the problem is that even in that case, the vocabulary, i.e. the set of words that can be embedded may not cover all the words for some downstream tasks. One solution is to train the Word2Vec on the text data for the task at hand which, however, are not always large enough to build a good model. In our study, we train our model since the size of our corpus is moderately large.

Another way to transform words into vectors is to use the so-called embedding matrix. Each word representation as the one-hot vector is multiplied by this matrix to produce the desired word embedding vector. Vectors from all the words in the sentence are then fed to some neural network model and its parameters are trained jointly with the matrix weights. In our study, we used feed-forward (Linear) or LSTM recurrent network layer(s) which is designed to produce a single vector representing the whole sentence. This approach is schematically shown in Fig. 3.

2.2 Direct sentence embedding methods

Since the introduction of large-scale language models based on the transformer architecture such as BERT [6], many NLP tasks have seen a big boost in the performance. Trained on a large amount of text data collected from the Internet, BERT is capable of extracting information from the input text that older language models could not. The main advantage of the transformer-based models is the use of a multi-head self-attention mechanism coupled with deep multi-layer architecture. Input words are first tokenized using a model-specific tokenizer. It contains several special tokens, such as [CLS] and [SEP] which are used at the beginning of each input

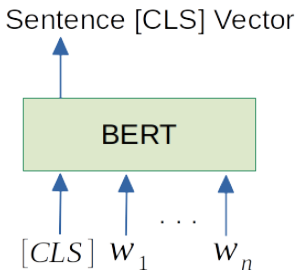

Figure 4: Sentence embedding using BERT's [CLS] token

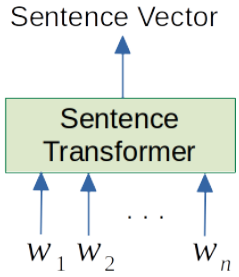

Figure 5: Sentence embedding using Sentence Transformer model

word sequence and between sentences if there is more than one. Because of the self-attention mechanism, the output corresponding to the [CLS] token holds information about all the input words and, thus can be used as a representation of the whole sentence. This is shown in Fig. 4 and is the first method we use for direct sentence embedding.

Recently, an improved version of BERT, specially designed for sentence embedding called Sentence Transformer [13] was introduced. The model structure is similar to BERT, but the training scheme is different. Two BERT models are coupled in a Siamese network and trained using two separate sentences with known relations: contradiction, entailment, and neutral. This improves the model performance in both classification and regression tasks. We use the Sentence Transformer in a way shown in Fig. 5.

Although BERT is trained on a large amount of training data, for some specific tasks such as emotion recognition, it is often beneficial to fine-tune it on in-domain data, i.e. data especially collected for that task. It is possible to use the dataset we utilized in this study for fine-tuning BERT, but that would make specific for this dataset and the performance could be biased. A better approach is to use different in-domain datasets for fine-tuning and evaluation. In our system, we used the BERT model fine-tuned on the emotion classification task with data from several other publicly available databases. The sentence embedding procedure, in this case, is the same as in Fig. 4 and we refer to this model as "fine-tuned BERT".

3 EXPERIMENTS

3.1 Dataset

In this study, we use the EmoBank dataset [3], a corpus of 10k multi-genre English sentences manually annotated with dimensional sentiment metadata in the Valence-Arousal-Dominance (VAD) representation format. It has a bi-perspectival design, i.e. labels are provided for both the writer's and reader's emotions. Each sentence's sentiment data consists of integer scores in the range from 1 to 5 for each VAD dimension.

In our experiments, we used only the Valence-Arousal (VA) labels of the reader's perspective. The dataset is divided into a training set of 8062 sentences, a development set of 1000 sentences, and a test set of 1000 sentences.

Table 1: Evaluation results for Valence prediction scores using word embeddings and Feed-Forward DNN back-end.

Sentence Embedding Method	Aggregation method			
	Linear		LSTM	
	R2	MAE	R2	MAE
Emb. Matrix	0.0116	0.2458	0.0671	0.2429
Word2Vec	0.0015	0.2459	0.0052	0.2466

3.2 Evaluation metrics

For regression tasks, there exist several evaluation metrics, such as the coefficient of determination (R^2), root mean squared error (RMSE) or mean absolute error (MAE). Each metric estimated the prediction performance from a different point of view and is often used in conjunction with others. The R^2, for example, is a measure of goodness of fit. It is the proportion of variance in the dependent variable that is explained by the model and is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2} \quad (1)$$

where y_i and $\hat{y}_i$ are the true and and predicted output values, and $\overline{y}_i$ is the mean of the true output values. The better regression output fits the true data, the closer is R^2 to 1.

The RMSE and MAE metrics measure the closeness of the predicted outputs to their corresponding true values. They are calculated by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (3)$$

In our experiments, we use the R^2 and MAE as system evaluation metrics.

3.3 Results

First, we experimented with word embedding-based methods. As described in Section 2.1, we use Embedding Matrix and Word2Vec methods for obtaining word embedding vectors. After some preliminary experiments, we set the embedding vectors' dimension to 256. All the word vectors are then aggregated into a single sentence vector using either a Linear or recurrent (LSTM) network layer. The size of the sentence vectors is kept the same - 256. As for the back-end, in this case, only Feed-Forward DNN is feasible, since the whole system has to be trained end-to-end with the Back Propagation algorithm. The other back-end model, SVR, uses a different training algorithm and cannot be coupled with this front-end. Using the validation dataset we tuned the DNN back-end model hyperparameters such as the number of layers, layer size, activation function, learning rate, batch size, etc. The final model architecture is as follows: 3 layers with sizes 128, 32, and 1, ReLU activation and learning rate of 0.0005 batch size of 16. In Table 1 and Table 2, we provide the results in terms of R^2 and MAE for Valence and Arousal scores respectively.

Table 2: Evaluation results for Arousal prediction scores using word embeddings and Feed-Forward DNN back-end.

Sentence Embedding Method	Aggregation method			
	Linear		LSTM	
	R2	MAE	R2	MAE
Emb. Matrix	0.1108	0.1904	0.0082	0.1887
Word2Vec	0.0013	0.1898	0.0036	0.1909

Table 3: Evaluation results for Valence prediction scores using sentence embeddings and both the Feed-Forward DNN and SVR back-ends.

Sentence Embedding Method	Back-end			
	DNN		SVR	
	R2	MAE	R2	MAE
BERT	0.3359	0.2092	0.2791	0.2726
Sent.Transformer	0.4806	0.1932	0.4721	0.2671
Fine tuned BERT	0.0370	0.2683	0.5413	0.2660

As can be seen from these tables, both the Embedding Matrix and Word2Vec methods have very similar performance with either the Linear or LSTM-based word vector aggregation approaches. The low values of R^2 coefficient suggest almost no correlation. However, the more important metric for this task is the MAE which is about 0.24 and 0.19 for the Valence and Arousal respectively.

Next, we evaluated the direct sentence embedding methods described in Section 2.2. In all the cases we used pre-trained language models freely available on the Internet. There are several versions of the BERT model to choose from, so we selected the DistilBERT version which is s a small, fast, cheap, and light model trained by distilling BERT base. It has 40% fewer parameters and runs 60% faster while preserving over 95% of BERT's performance. The Sentence Transformer model we utilized was the default multilingual model from the "SentenceTransformers" package [12]. The last model we tried is a DistilBERT fine-tuned on the "GoEmotions" dataset. It is a corpus of 58k carefully curated comments extracted from Reddit, with human annotations to 27 emotion categories or Neutral. Although this dataset is designed for the emotion classification task, we expect that the fine-tuning will be beneficial for the regression task as well.

The output vectors of all BERT-based models have a size of 768, which in our case is the sentence embedding vector dimension. Thus, the structure of the DNN back-end regression model is 768-256-64-32-16-1 in terms of layer sizes. For the SVR back-end, we found that the rbf kernel is better and the best value of C is 1. We summarise the results we obtained with direct sentence embedding models in Table 3 and Table 4 for the Valence and Arousal scores respectively.

In terms of MAE, there is no clear winner among the different methods, but compared with the word level embedding approaches, direct sentence embedding is better, especially for the Valence prediction scores. Both the DNN and SVR back-ends achieve comparable results for Arousal scores, but the DNN is better for Valence. As for the R^2 metric, there is some variability in the performance of the BERT-based methods, however, their goodness of fit, i.e. the

Table 4: Evaluation results for Arousal prediction scores using sentence embeddings and both the Feed-Forward DNN and SVR back-ends.

Sentence Embedding Method	Back-end			
	DNN		SVR	
	R2	MAE	R2	MAE
BERT	0.0903	0.1748	0.0542	0.1867
Sent.Transformer	0.2017	0.1814	0.1138	0.1808
Fine tuned BERT	0.0858	0.1854	0.2163	0.1731

R^2 values are much higher than those from the word embedding approaches.

4 CONCLUSION

In this study, we investigated the performance of several word and sentence level text embedding methods for the task of predicting emotion Valence and Arousal scores. To assure a fair comparison, we used the same regression models as back-ends such as Feed-Forward DNN and SVR.

At the word-level, we applied Embedding Matrix and Word2Vec models and used Linear or LSTM layers to aggregate word vectors into a single sentence embedding vector. On the other hand, language models based on the BERT architecture can produce sentence-level embedding directly. Our experiments involved three different BERT versions: DistilBERT, SentenceTransformer, and DistilBERT fine-tuned on the GoEmotion dataset.

We compared the performance of all sentence embedding methods in terms of R^2 and MAE metrics using the EmoBank database. The results show that on average, the BERT-based approaches perform better than word-level embeddings, which is expected because the language models are trained on a much larger amount of data despite being from different domains. Fine-tuning on in-domain data seems to be helpful, but not significantly. To confirm the obtained results, we plan to experiment with other emotional databases as well.

REFERENCES

[1] Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of BERT-based approaches. *Artificial Intelligence Review* 54, 8 (2021), 5789–5829.

[2] Nourah Alswaidan and Mohamed El Bachir Menai. 2020. A survey of state-of-the-art approaches for emotion recognition in text. *Knowledge and Information Systems* 62, 8 (2020), 2937–2987.

[3] Sven Buechel and Udo Hahn. 2022. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. *arXiv preprint arXiv:2205.01996* (2022).

[4] Hua Cai and Yingzi Lin. 2011. Modeling of operators' emotion and task performance in a virtual driving environment. *International Journal of Human-Computer Studies* 69, 9 (2011), 571–586.

[5] Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco. 2017. Affective computing and sentiment analysis. In *A practical guide to sentiment analysis*. Springer, 1–10.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Andrius Dzedzickis, Artūras Kaklauskas, and Vytautas Bucinskas. 2020. Human emotion recognition: Review of sensors and methods. *Sensors* 20, 3 (2020), 592.

[8] Pranav Goel, Devang Kulshreshtha, Prayas Jain, and Kaushal Kumar Shukla. 2017. Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

ACL, 58–65.
[9] Bernhard Kratzwald, Suzana Ilić, Mathias Kraus, Stefan Feuerriegel, and Helmut Prendinger. 2018. Deep learning for affective computing: Text-based emotion recognition in decision support. *Decision Support Systems* 115 (2018), 24–35.
[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).
[11] Saif M Mohammad. 2012. From once upon a time to happily ever after: Tracking emotions in mail and books. *Decision Support Systems* 53, 4 (2012), 730–741.
[12] Nils Reimers. [n. d.]. SentenceTransformers Documentation. www.sbert.net. ([n. d.]).
[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).
[14] James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology* 39, 6 (1980), 1161.
[15] Kashfia Sailunaz, Manmeet Dhaliwal, Jon Rokne, and Reda Alhajj. 2018. Emotion detection from text and speech: a survey. *Social Network Analysis and Mining* 8, 1 (2018), 1–26.
[16] Anvita Saxena, Ashish Khanna, and Deepak Gupta. 2020. Emotion recognition and detection methods: A comprehensive survey. *Journal of Artificial Intelligence and Systems* 2, 1 (2020), 53–79.
[17] Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of language and social psychology* 29, 1 (2010), 24–54.
[18] Samira Zad, Maryam Heidari, James H Jr Jones, and Ozlem Uzuner. 2021. Emotion Detection of Textual Data: An Interdisciplinary Survey. In *2021 IEEE World AI IoT Congress (AIIoT)*. 0255–0261. https://doi.org/10.1109/AIIoT52608.2021.9454192