Embedding Text in Emotion and Evocation Vector Spaces

Anonymous ACL submission

Abstract

The aim of this paper is to present the methodology for embedding text in a concrete emotion vector space, where each dimension represents a single emotion, and the coordinates represent emotion intensities. Additionally, the text is also embedded in an experimental (emotion) evocation vector space, where the coordinates represent possible evocation intensities of each emotion for the text reader. Embeddings are performed using newly prepared sentence transformer-based language models, trained on an existing dataset of social media posts written in Serbian and manually labeled with emotions.

1 Introduction

In the last decade, we have seen a steady increase in interest for the domain of sentiment and emotion analysis, with the boundaries of emotion classification research mostly being pushed by the industry and the need for better understanding of customer satisfaction and for providing similar insights. Traditionally, this task was handled by employing machine learning techniques over a labeled dataset (Alm et al., 2005; Liu et al., 2023) or by fine-tuning various pre-trained language models on emotion classification task (Adoma et al., 2020). These traditional emotion classification methods output binary results (both in case of single-label and multi-label classifications), meaning an emotion is either detected or not detected, leading to loss of a more nuanced emotional content, for example, emotion intensities. Although we can build a more value-specific dataset (Mohammad and Bravo-Marquez, 2017) or use scores based on label probabilities to aid in solving this issue, a more straight-forward solution would be to directly embed the text in the emotion vector space, enabling the quantification of emotion intensity across multiple dimensions.

This paper dives into the building of such embedding models, by training them on a text-labeled dataset of social media posts in Serbian. Additionally, it explores the concept of emotion evocation via embedding the text in a vector space that reflects potential emotional responses elicited in readers. The dual embedding spaces, emotion and emotion evocation, aim to provide a more comprehensive representation of both emotions expressed in text and its potential emotional impact.

2 Dataset

The base dataset used for this research, *SocialEmo.Sr* (Šošić et al., 2024) consists of 16,670 *X* (*Twitter*), and 17,930 *Reddit* posts written in Serbian and manually annotated for one or more of the Plutchik's eight basic emotions (joy, trust, fear, surprise, sadness, anticipation, anger, and disgust (Plutchik, 1965)), by multiple human annotators. To create the final training set of text-vector pairs, the following columns were used:

id – The id of the specific row, derived from the *conversation_id* (*Twitter*) or *submission_id* (*Reddit*).

text_lat – Textual content of the post;

gold_label – A list of emotion labels, consolidated from multiple annotator inputs.

referenced_tweets_id – A list of id-s *Twitter* post is a reference to.

parent_id – An id of the post *Reddit* post is a reference to.

For each reference made using *referenced_tweets_id* or *parent_id*, both emotion and emotion evocation embeddings for the text in *text_lat* (from referenced post) were produced using the assigned *gold_labels*. Emotion embeddings were assigned using labels from the referenced post, and evocation embeddings were produced using the labels from the referee post. This is

done under the assumption that the referenced post evoked emotions present in the referee post.

All of the textual labels were mapped to integer values representing emotion dimension indices, so, given a set of mapped gold labels from the referenced post $L^1 \subseteq \{0..7\}$, and a set of mapped gold labels from the referee post $L^2 \subseteq \{0..7\}$ resulting embedding vector v is defined as follows:

$$v_i = \begin{cases} 1 & \text{if } i \in \{L^1\} \text{ or } i-8 \in \{L^2\} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i \in \{0..15\} \tag{1}$$

The transformation resulted in a set of 15,412 unique pairs of texts and 16-dimension embeddings, 8 for emotions and 8 for emotion evocations. This set was then shuffled and split using a 8-1-1 ratio to get training, validation and test splits for the experiment.

3 Models

In order to build combined models for emotion and evocation embedding, two approaches were envisioned, both using the *sentence transformers* architecture (Reimers and Gurevych, 2019) and *Jerteh355* (Škorić, 2024) Serbian text embedding language model (355 million parameters) as the base word-embedding model.

Since sentence transformers are typically trained using text triplets or pairs with similarity values, only two of its loss functions support vector inputs: *MSELoss* (Mean Squared Error Loss) and *Margin-MSELoss*, both intended for knowledge distillation from a *teacher* model to a *student* model of the same embedding dimension. We can, however, set the student model's embedding dimension to match the set of possible labels, and provide the *teaching* embeddings directly from the prepared dataset.

3.1 Single Emotion and Evocation Model

The first approach employed a simple sentence transformer architecture, consisting of the base word-embedding model, one mean pooling layer and one dense layer with 16 outputs: 8 for emotions and 8 for emotion evocations (Figure 1).

The training of the model was conducted using the prepared training and validation splits and default hyperparameters (batch size 8, learning rate of 5e-5 etc.) for one epoch, which resulted in a model that embeds text in 16 dimensions, where the first 8 are for the expressed emotions intensity and the latter 8 for the intensity of emotions the text possibly evokes.

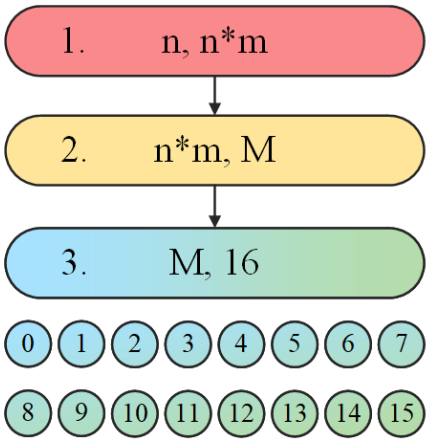

Figure 1: Architecture of a singular emotion and evocation embedding model, where *n* is the number of input words, *m* is the word-embedding dimension (1024) and *M* is the pooling output dimension (also 1024). Color red represents the word-embedding model, yellow the mean-pooling layer and blue-green represents the dense layer for emotion and evocation embedding.

3.2 Combined Emotion and Evocation Model

The second approach also employed the sentence transformer architecture but involved training of two separate models that are later merged: one for embedding in emotion space and another for embedding in evocation space.

Just like in the first example, each model consisted out of the base transformer model, a mean pooling layer, and a dense layer, except that the dense layers had an output dimension of 8. For each row in the prepared dataset, the first 8 dimensions of the embedding vector were used to train the emotion embedding model, while the later 8 were used to train evocation embedding model.

While this approach allows for the usage of separate training sets for each respective model, their latter merge comes with a constraint: the word embedding model weights must be frozen for the training of at least one model. In order to minimize this loss, we can perform the full training for the emotion embedding model, and then fine tune that model with evocation embedding dataset while freezing the parameters in the word-embedding model to secure that the output dense layers receive the same input for both models, enabling their merge into a new, combined, sentence transformer model (Figure 2).

This model has the following components:

1. A single word-embedding model, which can be copied from either initial model.

2. A single mean pooling layer with the same output dimension as those in the initial models.

3. A new *mapping* dense layer, used to duplicate the output of the pooling layer, as the new embedding layer's input dimension is doubled in size. This is accomplished by initializing a dense layer without bias, with the input dimension matching the output dimension of the pooling layer, and the output dimension being twice as big. To ensure the output of this layer is equal to two concatenated replicas of the input, the weights are set as follows:

$$w_{ij} = \begin{cases} 1 & \text{if } j-i \in \{0, n-1\} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$
$$\text{for } i \in \{0..n-1\}, j \in \{0..2n-1\}$$

where w is weight, i is the input index, j is the output index, and n is the input dimension.[1] Also, it is necessary eliminate any activation functions that could corrupt the value pass-through.[2]

4. Finally, a new embedding dense layer, with an input dimension double the size of the pooling layer output and the output dimension of 16, which aims to encapsulate the embedding layer weights from both initial models. To accomplish that, the weights and bias for this layer are set as follows:

$$w_{ij} = \begin{cases} w^1_{ij} & \text{if } i < n \text{ and } j < 8 \\ w^2_{i-n,j-8} & \text{if } i \geq n \text{ and } j \geq 8 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$
$$\text{for } i \in \{0..2n-1\}, j \in \{0..15\}$$

$$b_j = \begin{cases} b^1_j & \text{if } j < 8 \\ b^2_{j-8} & \text{otherwise} \end{cases} \quad (4)$$
$$\text{for } j \in \{0..15\}$$

[1]The same can be accomplished by creating a custom forwarding function, but that would require later users to enable *trust remote code* parameter in order to run the model, which is not advisable. This approach bypasses that issue.

[2]Sentence transformers dense layers use Hyperbolic Tangent (Tanh) activation by default, and there is no option for linear activation, since it does not exist in the reference module, *PyTorch*. We can, however, still perform the pass-through by *planting* PyTorch placeholder, argument-insensitive operator, *nn.Identity*, as the activation function for this layer.

where w and b are the weights and bias in the new dense layer, w^1 and b^1 are the weights and bias of the embedding layer from the first pre-trained model, w^2 and b^2 are the weights and bias of the embedding layer from the second pre-trained model, i is the input index, j is the output index, and n is the original input dimension.

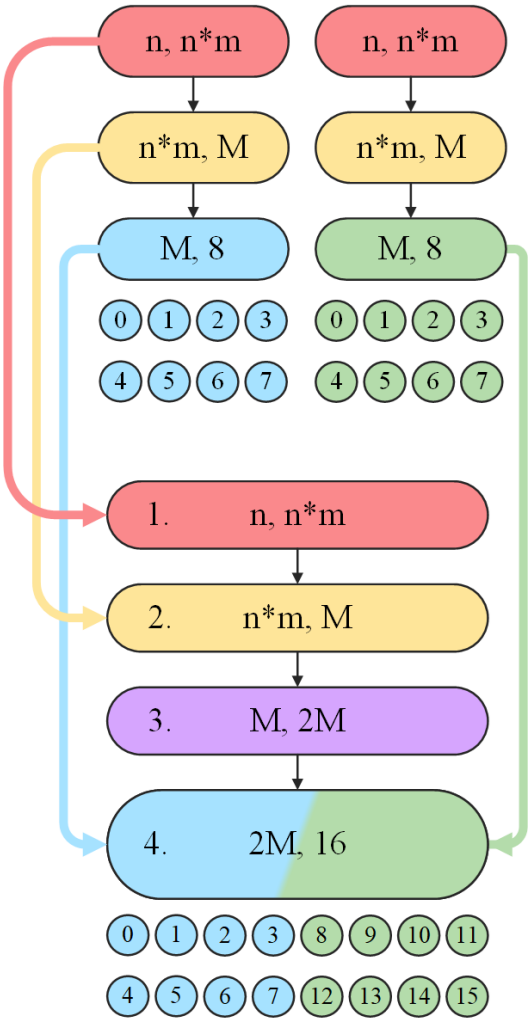

Figure 2: Architecture of a combined emotion and evocation embedding model, where *n* is the number of input words, *m* is the word-embedding dimension (1024) and *M* is the pooling output dimension (1024). Word-embedding modules are depicted in red, mean-pooling layers in yellow, dense layers for emotion embedding in blue and dense layers for evocation embedding in green. Purple represents the mapping dense layer. Colored arrows represent layer origins for the final model (bottom) from the initial emotion (top left) and evocation (top right) embedding models.

Just like the single embedding model (section 3.1), this model outputs a single embedding vector with a dimension of 16, where 8 values represent

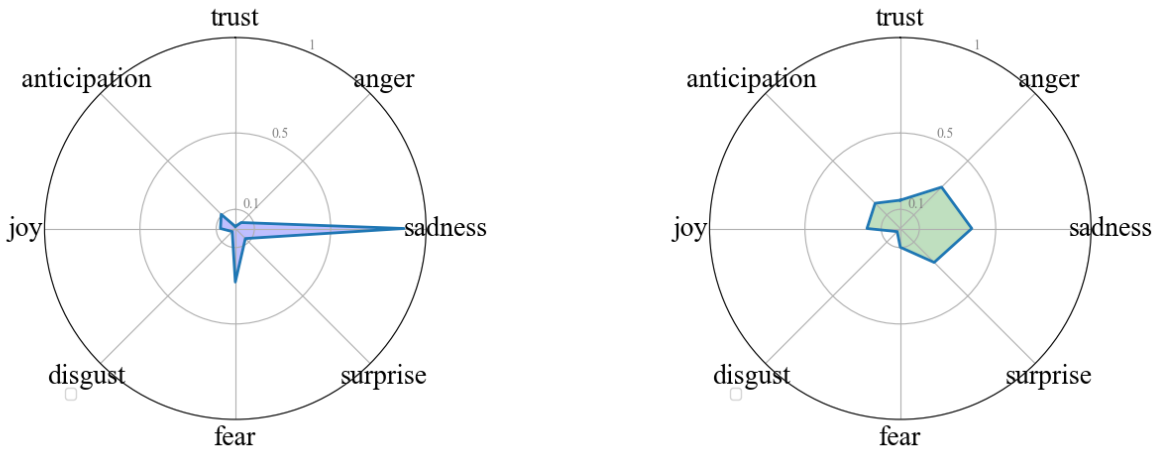

Figure 3: *There's a lot of traffic, so I'll be a little late.* – Difference between calculated emotion (left) and emotion evocation intensities (right).

the intensity of emotions expressed in the text and 8 represent the emotion evocation intensity. An example of the outputs is visualized in Figure 3.

4 Discussion

The paper details the construction of emotion and emotion evocation embedding vector spaces through training of a series of sentence-transformer models.

Two different approaches for the creation of these dual (emotion and emotion evocation) embeddings are presented, one where both embedding spaces are entangled during training, and the other where those spaces are separated. When testing the mean square error loss of these models against the test dataset, we find that the second model is slightly better for embedding emotions (0.0779 to 0.0822), but it is significantly worse for embedding emotion evocations (0.1800 to 0.1441). These findings come at no surprise, since the first model was fully trained for evocation embedding and the second one was merely fine-tuned. Disadvantage of the fine-tuning approach is even clearer when we calculate the loss between the test set emotion embeddings and the supposed evocation embeddings using a second model (0.0867), meaning the model is still embedding text in the original, emotion, rather than the fine-tuned evocation space. If we subject the first model to this test, we get a value of 0.1262, meaning the spaces are further apart.

All in all, the paper demonstrate that it is possible to build separate emotion and emotion evocation vector spaces for text embedding, enhancing the granularity of emotion analysis, and hopefully paving the way for improved sentiment analysis applications and emotionally intelligent systems. The embedding of text in these specialized vector spaces offers a powerful tool for researchers and practitioners to dissect and better understand the complex emotional landscape of written language. Finally, it shows that the models can be taught to produce granular emotional intensities from a non-granular dataset.

The discrepancy between emotion and evocation spaces also indicates the possibility of learning emotion evocation embeddings through the emotions of social media post references, which is still largely unexplored and could have a potential applicative impact in various fields: social media monitoring and marketing, psychological studies or even human-computer interaction.

Limitations

There are several limitations inherent to this study that should be acknowledged:

Training Set Configuration: The study opted for a simpler approach by not utilizing the obvious advantage of separate training sets for different tasks in the second approach. This decision was made to avoid overcomplicating the paper. However, the inclusion of non-referenced posts in the training set for the second could have potentially impacted the results. Additionally, the strict limit in the number of fine-tuning epochs for the evocation embedding model may have further negatively influenced the model's performance.

Evaluation Limitations: The evaluation of the embedding model against the test set presents some

challenges. Given the differences in granularity, it is uncertain whether this evaluation method is entirely sufficient. Typically, the performance of embedding models is assessed through downstream tasks, which was beyond the scope of this paper, thus, the evaluation provided here may not be fully conclusive. Extensive evaluation using a variety of downstream tasks is necessary to better understand the model's capabilities.

Model usability: The models developed in this study were built using experimental methods on a single dataset, not previously established or thoroughly super-evaluated. Further validation on more diverse and well-established datasets is necessary to confirm the robustness and applicability of these models in real-world scenarios.

References

Acheampong Francisca Adoma, Nunoo-Mensah Henry, and Wenyu Chen. 2020. Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 117–121. IEEE.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 579–586, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Xuan Liu, Tianyi Shi, Guohui Zhou, Mingzhe Liu, Zhengtong Yin, Lirong Yin, and Wenfeng Zheng. 2023. Emotion classification for short texts: an improved multi-label method. *Humanities and Social Sciences Communications*, 10(1):1–9.

Saif Mohammad and Felipe Bravo-Marquez. 2017. WASSA-2017 shared task on emotion intensity. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 34–49, Copenhagen, Denmark. Association for Computational Linguistics.

Robert Plutchik. 1965. What is an emotion? *The Journal of psychology*, 61(2):295–303.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Mihailo Škorić. 2024. Novi jezički modeli za srpski jezik. *Preprint*, arXiv:2402.14379.

Milena Šošić, Ranka Stanković, and Jelena Graovac. 2024. Social-emo. sr: Emotional multi-label categorization of conversational messages from social networks x and reddit. In *South Slavic Languages in the Digital Environment JuDig Book of Abstracts, University of Belgrade-Faculty of Philology*, page 58.