TILeN at SemEval-2025 Task 11: A Transformer-Based Model for Sentiment Classification Applied to the Russian Language

Jorge Reyes-Magaña
Universidad Autónoma de Yucatán
Facultad de Matemáticas
jorge.reyes@correo.uady.mx

Luis Basto-Díaz
Universidad Autónoma de Yucatán
Facultad de Matemáticas
luis.basto@correo.uady.mx

Luis Fernando Curi-Quintal
Universidad Autónoma de Yucatán
Facultad de Matemáticas
cquintal@correo.uady.mx

Abstract

We present our approach to tackle the Sentiment Classification Task. The task was divided into 3 categories: 1) Track A: Multi-label Emotion Detection 2) Track B: Emotion Intensity, and 3) Cross-lingual Emotion Detection. We participate in subtasks 1 and 2 for the Russian language. Our main approach is summarized as using pre-trained language models and afterwords working with fine-tuning aside the corpora provided. During the development phase, we had promising outcomes. Later during the test phase, we got similar scores to the Semeval baseline. Our approach is easy to replicate and we proportionate every detail of the process performed.

1 Introduction

Nowadays, a significant portion of human communication takes place through text-based platforms such as social media, emails, and messaging applications. Understanding the emotions embedded in these interactions is crucial for enhancing user experience, analyzing public opinion, and even detecting mental health issues. Text-Based Emotion Detection (TBED) is a field within Natural Language Processing (NLP) that aims to identify and classify emotions in written text using machine learning and artificial intelligence techniques. This technology has diverse applications, ranging from sentiment analysis in social media to personalized content recommendations and customer service improvement. However, accurately detecting emotions in text remains a challenging task due to the inherent ambiguity of language, subjectivity in emotional expression, and cultural differences (Alswaidan and Menai, 2020).

SemEval-2025 Task 11 seeks to identify the perceived emotion most people would associate with a speaker based on a given sentence or short text snippet. This task prioritizes interpreting emotions rather than the speaker's actual emotional state, the emotions elicited in the reader, or those of other individuals referenced in the text. Its significance lies in enhancing the understanding of how emotions are conveyed and perceived in written language, considering the influence of cultural context, individual variations in emotional expression, and the inherent constraints of text-based communication (Muhammad et al., 2025b). This task consists of three tracks:

- Track A: Multi-label emotion detection
- Track B: Emotion intensity
- Track C: Cross-lingual emotion detection

TILeN group participated in tracks A and B for the Russian language, using in both tracks the pre-trained model RuBERT (Kuratov and Arkhipov, 2019) fine-tuned with the task data (Muhammad et al., 2025a). Due to the intensity of the classes for emotions included in track B, we pre-processed the corpora, transforming all classes into a binary classification with three intensity levels for emotions applying multi-label classification.

In the final ranking, our results were below the SemEval base score in both tracks, but the differences were within hundredths of a unit. Our approach demonstrated improved performance when employing a language-specific pre-trained model for multi-label classification predictions.

2 Related work

2.1 BERT Model

Language models pre-training has been shown to be effective for improving many natural language processing tasks, such as, sentence-level task (Williams et al., 2018) and paraphrasis (Dolan and Brockett, 2005).

A downstream task depends on the output of a previous task, and it allow us to use pre-trained models for various applications. Two strategies

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 2044–2049
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

for applying pre-trained language representations to downstream tasks are: feature-based and fine-tuning.

- The feature-based strategy uses the pre-trained representations as additional features, such as ELMo (Peters et al., 2018).

- The fine-tuning is trained on the downstream tasks by simply fine-tuning all pre-trained parameters such as Generative Pre-trained Transformer (OpenAI GPT) (Radford and Narasimhan, 2018).

The two approaches use unidirectional language models to learn general language representations during pre-training and this limits the choice of architectures. Here is where BERT model comes in.

BERT which stands for Bidirectional Encoder Representation from Transformers is a natural language processing model based on the Transformer architecture. It was developed by Google AI in 2018 and revolutionized the field of NLP because it allows bidirectional learning of texts, which improves understanding of context in both directions (left and right).

BERT alleviates the unidirectionality constraint and use two steps, pre-training, and fine-tuning. In the pre-training step, the model is trained on unlabeled data over different pre-training task, on the other hand, for fine-tuning, the BERT model is first initialized with pre-trained parameters, and all of them are fine-tuned using labeled data from the downstream task (Devlin et al., 2019). BERT shows state-of-the-art results on a wide range of NLP tasks in English.

BERT has two multilingual models currently available.

- BERT-Base, Multilingual Cased: with 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters

- BERT-Base, Chinese: Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters

2.1.1 BERT Architecture

Before a text is fed into the model, it must be split into tokens. BERT uses a special tokenization called WordPiece (Wu et al., 2016), which splits words into subwords or fragments.

The BERT's architecture is described below

The Transformer Network processes the text in two phases: one for encoding, which is responsible for processing the input text and numerically encoding it by extracting its most relevant information, and a decoding phase that is responsible for generating a new text sequence.

The input text is encoded as two sentences:

At the beginning, a classification token (CLS) is always included and to separate one sentence from another, the separation token (SEP) is included. For each token, three representations are obtained:

- Token Embeddings: Representation of each word or subword.

- Segment Embeddings: Indicate which sentence each token belongs to.

- Positional Embeddings: Add information about the position of each token in the sequence.

BERT is trained with two main goals:

- Masked Language Model (MLM): Some tokens in the input are randomly masked and the model must predict them.

- Next Sentence Prediction (NSP): Two sentences are given and the model must predict whether the second sentence follows the first.

2.2 RuBERT Model

There are two ways to train pre-trained bidirectional language models monolingual and multilingual. Language specific models have shown greater performance than multilingual models, but these last ones allow to perform a transfer from one language to another and solve task for different language simultaneously.

This work (Kuratov and Arkhipov, 2019) shows several methods of adaptation of multilingual masked language models for a specific language, it's a Transformer encoder where the basic building blocks are Self-Attention.

This model was trained on the Masked Language Modeling and next sentence prediction tasks and considered from multilingual to monolingual using Russian as a target language for transfer. It demonstrated that the monolingual model could be trained using multilingual initialization.

The main idea is to use knowledge about target language that already captured during multilingual training. So, training model using data from multiple languages can significantly improve the performance of the model.

3 System Overview

We participated in Tracks, A and B for the Russian Language since we got promising results during the development phase. Both tracks were tackled using the same approach. We used the pretrainned model RuBERT based on Bert for the Russian Language (Kuratov and Arkhipov, 2019). This model is cased and it was trained with 12-layer, 768-hidden, 12-heads and 180m parameters. The corpora used for training consist of the Russian part of Wikipedia and news data. The authors used the training data to build a vocabulary of Russian subtokens and took a multilingual version of BERT-base as an initialization for RuBERT.

After loading the pre-trained model, we fine-tuned it using the task data (Muhammad et al., 2025a) provided by the organizers.

3.1 Track A. Multi-label Emotion Detection

This task is intended to predict the perceived emotion(s) of the speaker given a target text. The emotions to detect are: joy, sadness, fear, anger, surprise, or disgust. In other words, label the text snippet with the emotion (1) or the absence of it (0). In the task it's possible to have the presence of two or more emotions in the same text.

For example, the text: *You know what happens when I get one of these stupid ideas in my head.* Is labeled as anger and fear in the training corpus.

Additionally, the task includes a large number of languages with many predominantly spoken in regions characterized by a relatively limited availability of NLP resources (e.g., Africa, Asia, Eastern Europe and Latin America): Afrikaans, Algerian Arabic, Amharic, Hausa, Igbo, Kinyarwanda, Moroccan Arabic, Mozambican Portuguese, Nigerian-Pidgin, Oromo, Setswana, Somali, Swahili, Tigrinya,Xitsonga, isiXhosa, Yoruba, isiZulu Arabic, Chinese, Hindi, Indonesian, Javanese, Marathi English, German, Romanian, Russian, Latin American Spanish, Tatar, Ukrainian, Swedish Brazilian Portuguese

Our team only participates in the predictions of texts written in Russian. The process used is described on algorithm 1. To obtain a larger amount of data we joined the training and development corpora. So we could have more text which is essential for this approach. This algorithm was devised as a simple way to achieve good results in multi label classification.

Algorithm 1 Training a BERT-based model for multi-label sentiment classification

```
Require: Corpora with texts and binarized labels, pretrained RuBERT model
Require: Learning rate $\eta = 1e - 5$, batch size $B = 2$, epochs $E = 10$
Load and preprocess data
Read dataset and convert labels to multi-label binary format
Tokenize texts using BertTokenizer (max length = 128)
Create DataLoader with batch size $B$
Initialize model and training setup
Load BertForSequenceClassification with $num\_labels = |labels|$
Define Adam optimizer and BCEWithLogitsLoss for multi-label classification
Train model for $E = 10$ epochs
for $epoch = 1$ to $10$ do
    for each batch in DataLoader do
        Perform forward pass and compute logits
        Compute loss and update model weights via backpropagation
    end for
    Print epoch loss
end for
```

Due to hardware restrictions we couldn't add more epochs to our training. We didn't perform any additional pre-process to the corpora provided, so in this manner the data used were introduced to the algorithm without any changes.

3.2 Track B: Emotion Intensity

This track is designed to predict the intensity of each emotion class. Given a target text and a target perceived emotion.

The set of perceived emotions is the same for Track A, including: joy, sadness, fear, anger, surprise, or disgust.

The set of intensity classes are:

- 0: No emotion
- 1: Low degree of emotion

- 2: Moderate degree of emotion
- 3: High degree of emotion

Additionally, Track B also contains text snippets that could have multi-degree sentiment classes. For example:

I can't believe it! I won the scholarship! This is amazing!

Having a value of 3 for joy and a value of 3 for surprise, i.e., high degrees of joy and surprise.

The process used to predict Track B was the same as used in Track A, with some additional pre-processing in the corpora. Due to the intensity classes in this Track B, we didn't have binary values assigned to each text snippets. However, we transform all the classes into a binary classification task. All the sentiment class were divided into 3 different classes. For example, the class joy was transformed into joy_low, joy_med, and joy_hig. Therefore, if the joy class value was set to 3, only the class joy_hig was set to 1 and the other two had 0. The same logic was applied to all sentiments and emotion degrees in the corpora. The algorithm 2 describes the transform we made to pre-process all the corpora used in this Track.

Algorithm 2 Preprocessing Emotion Labels in a Sentiment Dataset

```
Require: CSV file containing text data with emotion intensity levels (1 = low, 2 = medium, 3 = high)
Load dataset from CSV file into a dataframe $df$
Define emotion labels: {"anger", "disgust", "fear", "joy", "sadness", "surprise"}
Transform emotion levels
for each emotion $e$ in the emotion list do
    Create new columns:
        $e\_low \leftarrow 1$ if $e = 1$, else 0
        $e\_med \leftarrow 1$ if $e = 2$, else 0
        $e\_hig \leftarrow 1$ if $e = 3$, else 0
end for
Select relevant columns
Keep only {"id", "text"} and transformed emotion columns
Save processed dataset as a new CSV file
```

Finally, after performing the prediction for this Track, we need to return the corpora to the original distribution format. As described in the Algorithm 3.

Algorithm 3 Transform Emotion Labels in a Sentiment Dataset

```
Require: CSV file with separate columns for low, medium, and high emotion intensity
Load dataset from input CSV file into a dataframe $df$
Define emotion labels: {"anger", "disgust", "fear", "joy", "sadness", "surprise"}
Aggregate emotion intensity levels
for each emotion $e$ in the emotion list do
    Compute aggregated emotion value:
        $e \leftarrow e\_low \times 1 + e\_med \times 2 + e\_hig \times 3$
    Clip values to a maximum of 3
end for
Select final columns
Keep only {"id"} and the transformed emotion columns
Save transformed dataset to output CSV file
```

4 Experimental Setup

As mentioned in the previous section, we consider the new corpora for training the text snippets of train + dev to train our models.

- **Track A.** The training corpus used for this Track in Russian language consists of 2878 (2679 train / 199 dev) text snippets. Table 1 shows the classes distribution of the corpus.

Emotion	Count
Anger	590
Disgust	299
Fear	349
Joy	589
Sadness	460
Surprise	381

Table 1: Emotion distribution in dataset

Even though, the corpus it's not perfectly balanced, we can appreciate that all classes have at least 299 elements, which is helpful for the Algorithm 1 .

- **Track B.** Applying the same criteria to add train(2220)+ dev(343) we had a total of 2563 snippet texts with the sentiment intensity distribution as shown in Table 2.

Table 2 shows that emotion intensity 1 is the least represented of all emotions and the most is intensity 2.

Emotion	Emotion Intensity		
	1	2	3
Anger	28	218	171
Disgust	24	125	17
Fear	89	197	42
Joy	77	252	152
Sadness	33	215	86
Surprise	43	164	58

Table 2: Emotion distribution across intensity levels

The main Algorithm 1 is established in the previous section. We only made additional adjustments to the Track B corpus as mentioned before.

5 Results

After the Test stage, we had the following results.

- **Track A.**

Language	Emotion	Score
Russian	Macro F1	0.8209
	Micro F1	0.8196
	Anger	0.8131
	Disgust	0.8070
	Fear	**0.9254**
	Joy	0.8624
	Sadness	0.7589
	Surprise	0.7583

Table 3: Emotion classification scores for Russian

For the official ranking, organizers used the Macro F1 score and we were positioned in the 31st place on the final ranking for this task.

- **Track B.**

Language	Emotion	Score
Russian	Anger	0.7654
	Disgust	0.8053
	Fear	**0.8919**
	Joy	0.7678
	Sadness	0.8580
	Surprise	0.7540
	Average Pearson r	0.8071

Table 4: Emotion intensity classification scores for Russian

The average Pearson r was the score used by the organizers to rank official results. The position obtained for this Task was 17.

The model A exhibited rapid convergence throughout training. Starting with an initial loss of 0.1192 in Epoch 1, the loss quickly decreased to 0.0315 by Epoch 2 and further to 0.0199 in Epoch 3, demonstrating efficient early learning. Minor fluctuations were observed in subsequent epochs, but the overall trend remained strongly downward. Notably, after Epoch 5, the model consistently achieved very low loss values, reaching as low as 0.0054 by Epoch 10. These results highlight the model's ability to effectively capture the data distribution and maintain stable performance throughout the later stages of training. In Model B, the training loss shows a general trend of effective learning, especially in the early epochs. Initially, there is a significant decrease in loss from 0.15 to 0.08 by the second epoch and further to 0.018 by the third, indicating rapid model convergence at the start. Although some fluctuations are observed in later epochs (e.g., an increase at epoch 4 and a notable spike at epoch 7), the overall trend suggests that the model adapts well and corrects itself quickly, as seen by the drop to 0.031 at epoch 8 and reaching a near-zero loss of 0.0006 at epoch 9. The final loss at epoch 10 (0.062) remains low, demonstrating that the model maintains good performance and stability across training.

Both ranking results were positioned below the Semeval baseline scores. However, we were closer to this baseline in Track A. The distance from the baseline was short, i.e. 0.0168 for Track A and 0.0695 for Track B.

6 Conclusion

We present a simple way to make predictions having multilabel classification for sentiment analysis. The system approach, actually uses the same process applied to both Tasks, changes were made on the Track B corpus to be considered as a multi-label binary classification Task. The approach is based on a pre-trained language model for a specific idiom, and after performing some fine tuning we adjust the weights for the predictions. In this way, we consider that our approach is easy to replicate when you have some specific resources, especially on the pre-trained fact. Considering this, we tried to participate in some other languages and due to the low digital resources we couldn't get good results. This problem is also visible in the baseline results provided by the organizers, for example in Afrikaans language the baseline score is 0.3741.

Even though we couldn't overcome the Semeval baseline, the results weren't disappointing.

As a future work, we will perform additional aggregation to the corpus by applying paraphrasing, or even adding the Track B corpus to Track A, affirming the fact that all the intensity classes marked with values different from zero, means the presence of that sentiment as a value of 1 in Track A.

Acknowledgments

This research was carried out within the framework of project CONAHCYT CF-2023-G-64.

References

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*, 62:2937 – 2987.

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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yuri Kuratov and Mikhail Y. Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *CoRR*, abs/1905.07213.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine de Kock, Nirmal Surange, Daniela Teodorescu, Ibrahim Said Ahmad, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino D. M. A. Ali, Ilseyar Alimova, Vladimir Araujo, Nikolay Babakov, Naomi Baes, Ana-Maria Bucur, Andiswa Bukula, Guanqun Cao, Rodrigo Tufino Cardenas, Rendi Chevi, Chiamaka Ijeoma Chukwuneke, Alexandra Ciobotaru, Daryna Dementieva, Murja Sani Gadanya, Robert Geislinger, Bela Gipp, Oumaima Hourrane, Oana Ignat, Falalu Ibrahim Lawan, Rooweither Mabuya, Rahmad Mahendra, Vukosi Marivate, Andrew Piper, Alexander Panchenko, Charles Henrique Porto Ferreira, Vitaly Protasov, Samuel Rutunda, Manish Shrivastava, Aura Cristina Udrea, Lilian Diana Awuor Wanzare, Sophie Wu, Florian Valentin Wunderlich, Hanif Muhammad Zhafran, Tianhui Zhang, Yi Zhou, and Saif M. Mohammad. 2025a. Brighter: Bridging the gap in human-annotated textual emotion recognition datasets for 28 languages.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Seid Muhie Yimam, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine De Kock, Tadesse Destaw Belay, Ibrahim Said Ahmad, Nirmal Surange, Daniela Teodorescu, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino Ali, Vladimir Araujo, Abinew Ali Ayele, Oana Ignat, Alexander Panchenko, Yi Zhou, and Saif M. Mohammad. 2025b. SemEval task 11: Bridging the gap in text-based emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, Vienna, Austria. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.