PWEITINLP at SemEval-2024 Task 3: Two Step Emotion Cause Analysis

Sofiia Levchenko[1], 01155482@pw.edu.pl
Rafał Wolert[1], 01151705@pw.edu.pl
Piotr Andruszkiewicz[1,2], piotr.andruszkiewicz@pw.edu.pl
[1]Warsaw University of Technology
[2]Samsung Research Poland

Abstract

ECPE (emotion cause pair extraction) task was introduced to solve the shortcomings of ECE (emotion cause extraction). Models with sequential data processing abilities or complex architecture can be utilized to solve this task. Our contribution to solving **Subtask 1: Textual Emotion-Cause Pair Extraction in Conversations** defined in the **SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations** is to create a two-step solution to the ECPE task utilizing GPT-3 for emotion classification and SpanBERT for extracting the cause utterances.

1 Introduction

This paper introduces an approach for the emotion-cause extraction problem in dialogues. An emotion cause is defined and annotated in the given subtask as a textual span. Input to the model is a conversation containing the speaker and the text of each utterance. The model output should include all emotion-cause pairs, where each pair contains an emotion utterance along with its emotion category and the textual cause span in a specific cause utterance, e.g.(U3_Joy, U2_"You made up!").

Our contribution to solving **Subtask 1: Textual Emotion-Cause Pair Extraction in Conversations** defined in the **SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations** (Wang et al., 2024) is as follows: i) utilize GPT-3 for emotion classification, ii) utilize SpanBERT architecture to extract the cause utterances in dialogues as Q&A task, iii) contribute to solving the ECPE by finding its possible solutions in other NLP fields.

The task was separated into two parts - emotion classification, called subtask 1.1, and emotion-cause pair extraction, termed subtask 1.2. We have used GPT-3 and the SpanBERT model for these subtasks. In this paper, we also reflect on the results we got in the case of both subtasks, namely emotion classification and emotion-cause pair extraction. Our model, with one test entry, scored 9th in the competition.[1]

2 Related work

In recent years, many authors have suggested their approach to solving the ECPE task. Xia and Ding (2019) defined the ECPE task and proposed a two-step framework. First, independent multi-task learning (named Indep) consisting of BiLSTM modules and interactive multi-task learning (called Inter-EC for a model that uses emotion extraction to improve cause extraction and Inter-CE for a model that uses cause extraction to enhance emotion extraction) was used to extract a set of emotion cases and a set of cause clauses. Secondly, the sets were paired to yield a set of candidate emotion-cause pairs. Finally, a logistic model regression was used to filter the pairs. This two-step framework suffers from error propagation from the first step to the second step. Ding et al. (2020a) has also proposed a one-step framework that takes emotion-cause pairs as a 2D representation scheme with BiLSTM modules. These representations are forwarded into the 2D Transformer framework to capture pair interaction. Finally, binary classification is conducted to extract valid emotion-cause pairs. The new proposed framework outperforms the two-step framework by 7.6 percentage points of the F1 score. Regarding the joint framework, Ding et al. (2020b) have proposed a sliding window multi-label learning scheme named ECPE-MLL. It works on the assumption that all clauses in a document are emotion clauses, and an emotion-oriented sliding window is built centered on each emotion clause. In each window, the emotion clause extracts one or more of the corresponding cause clauses (the iterative synchronized multi-task learning (ISML) model is introduced to solve these subtasks). The results of this

[1] https://codalab.lisn.upsaclay.fr/competitions/16141#results

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1097–1105
June 20-21, 2024 ©2024 Association for Computational Linguistics

learning can be transformed into emotion-cause pairs. This approach serves an excellent advantage over the two-step framework proposed before. Chen et al. (2022) have proposed two alignment mechanisms with a model named A^2Net. Text documents are encoded with BERT and a partition filter network (PFN) to implement the first alignment mechanism: feature-task alignment to produce emotion-specific, cause-specific, and interaction features. The features are applied for EE (emotion and interaction features), CE (cause and interaction features), and ECPE tasks (all features). The inter-task alignment reduces then the inconsistency between label spaces among all tasks. The proposed framework achieves a higher F1 score and recall in the ECPE task, a higher F1 score in the EE task, and a higher recall and F1 score in terms of the CE task when compared to ECPE-2D.

3 Methodology

The emotion extraction cause task consists of two components - emotion extraction from the conversation and emotion cause span extraction. The first one could have been considered as a baseline for the second one, as we needed to identify which emotion and utterance should be used in the process of the cause search. There are two ways of approaching this problem. We could have created one model for both subtasks or separated it into two subsequent tasks, where each could be implemented using different models.

We have decided to go with the second approach, as we concluded that those less complex parts could have better quality in the end, even though we are aware of the error propagation, which definitely will be present in such a case.

3.1 Subtask 1.1

The first subtask aims to create a classification model, which will perform emotion recognition in each utterance of the conversation.

We have focused on two different models while approaching this problem. At first, we decided to use BiLSTM, but the results were not promising (refer to Appendix A and Section 4.1.2). Then, we have focused on utilizing the GPT-3 model (Brown et al., 2020) along with AssistantAPI provided by OpenAI.

3.1.1 Dataset

The training dataset, presented by the SemEval competition organizers, contained information about conversations between groups of people and emotion-cause pairs extracted from that conversation. The conversation consisted of multiple utterances, each with defined text, speaker, and emotion expressed by the speaker and their id.

The given dataset was transformed into a set of objects, where each represents a single utterance along with information about the context (concatenated utterances within the conversation) and expressed emotion.

3.1.2 Model

For the GPT-3 model, we have decided to use a standard Assistant (by only defining its purpose) and one enriched with data retrieval (by adding properly labeled data as its knowledge base).

The purpose of both Assistants was defined using the description:

*You are a system which analyzes conversation which consists of utterances sequence (attribute "context" in the given JSON object) among with given utterance (attribute "utterance" in the given JSON object) and then predicts emotion expressed (fear, surprise, joy, disgust, sadness, anger or neutral, you cannot use any other emotion as an answer and you must detect at least one of those emotions) in that utterance adding it to the answer using "**" symbol to emphasize answer's location.*

The enrichment of the second Assistant was based on the OpenAI *Knowledge Retrieval* functionality [2]. A selected number of records described further in Section 4.1 were fed into the GPT-3 model as a knowledge base. Upon querying, the model performs either a vector search or passes the file content to the context of the model calls. For further clearance, a model with/without a knowledge base will be called *GPT-3 based Assistant with/without knowledge base*.

3.2 Subtask 1.2

The second subtask aims to find which utterances in a given context are responsible for inducing the emotion predicted in subtask 1.1 (for details please refer to Section 3.1). The main idea of the second subtask is to fine-tune the SpanBERT model (Joshi et al., 2020) and perform question-answering to find the utterances in the dialogue for predicted emotion.

[2] https://platform.openai.com/docs/assistants/tools/knowledge-retrieval

Dataset	Count
`train`	5635
`validation`	1349
`test`	2380

Table 1: Train, validation, test dataset sizes for subtask 1.2

3.2.1 Dataset

The raw dataset that was presented in Section 3.1 and split into the 0.6-0.15-0.25 ratio was transformed to fit the question-answering task. The duplicates were removed. The original **text** field combined with the person speaking **speaker** in each **conversation** in the provided SemEval (Wang et al., 2023) dataset was converted into the **context** column. The question to be answered was formulated as follows: *What caused the [emotion]?*, where *[emotion]* refers to the predicted emotion for a given utterance combined into the context. Additionally, information was provided to indicate where the answer starts in the context, and **text** column to show the answer in the context. Table 1 shows the dataset sizes used for subtask 1.2.

A tokenizer was used with the original SpanBERT to fit the dataset into the SpanBERT input. Along the tokenization process, the following preprocessing steps were also applied:

1. For questions (column **question**) and contexts (column **contexts**), tokenization with truncation and padding on the right was applied. The max length of sequences was set to 512 (default SpanBERT value), and the stride was also used and set to 128 so that if the context is long, each of the features retrieved from the context has a context that overlaps the context from the previous feature.

2. For answers, the start position and end position were marked so that the current span's token index and the current span's end token index were put correctly even if the answer is out of span (CLS token was added in that case). If the answer was in a given span, the token start index and token end index were put to the two ends of the answer.

3.2.2 Model

The model used to finetune the data prepared for subtask 1.2 was SpanBERT[3] (Joshi et al., 2020) finetuned previously on the SQuAD v1.1 for the Q&A downstream task.

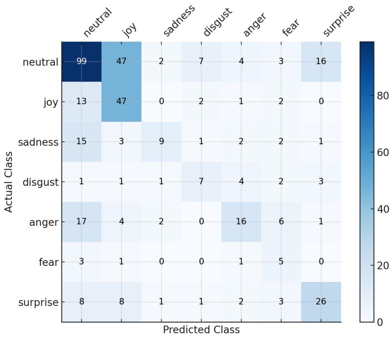

Figure 1: Confusion matrix for the GPT-3 based Assistant with knowledge base

4 Experimental Results

This section presents the experiment results (before evaluation phase) in terms of both subtasks.

4.1 Subtask 1.1

4.1.1 GPT-3 model

We have checked the accuracy of the GPT-3 model by utilizing 400 randomly selected utterances. Assistant, which was enhanced by adding a knowledge base, was using another randomly selected (but not similar to the ones in the test dataset) 500 records from the training dataset.

During the testing phase, we calculated the predicted labels' F1-score, accuracy, and recall and created a confusion matrix.

Figures 1, 2 and Tables 2, 3 refer to the confusion matrix for the Assistant with and without knowledge base accordingly.

Emotion	Precision	Recall	F1-Score
neutral	0.63	0.56	0.59
joy	0.42	0.72	0.53
sadness	0.60	0.27	0.37
disgust	0.39	0.37	0.38
anger	0.53	0.35	0.42
fear	0.22	0.50	0.30
surprise	0.55	0.53	0.54
Accuracy			0.52
Macro Avg	0.48	0.47	0.45
Weighted Avg	0.55	0.52	0.52

Table 2: Classification Report for the GPT-3 based Assistant with knowledge base

[3] HuggingFace implementation has been used.

Emotion	Precision	Recall	F1-Score
neutral	0.62	0.26	0.37
joy	0.31	0.86	0.46
sadness	0.27	0.09	0.14
disgust	0.32	0.37	0.34
anger	0.43	0.48	0.45
fear	0.25	0.50	0.33
surprise	0.41	0.35	0.38
Accuracy			0.39
Macro Avg	0.37	0.42	0.35
Weighted Avg	0.47	0.39	0.37

Table 3: Classification Report for the GPT-3 based Assistant without knowledge base

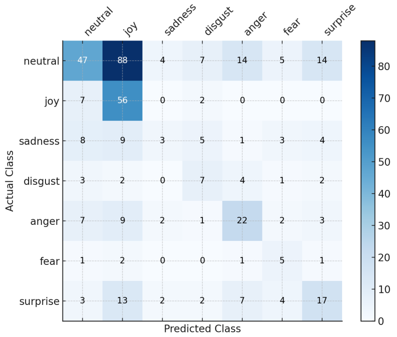

Figure 2: Confusion matrix for the GPT-3 based Assistant without knowledge base

Emotion	Precision	Recall	F1-Score
neutral	0.45	0.39	0.42
joy	0.24	0.35	0.28
sadness	0.14	0.09	0.11
disgust	0.06	0.03	0.04
anger	0.09	0.06	0.07
fear	0.05	0.04	0.04
surprise	0.20	0.32	0.25
Accuracy			0.28
Macro Avg	0.18	0.18	0.17
Weighted Avg	0.29	0.28	0.28

Table 4: Classification Report for Model 1

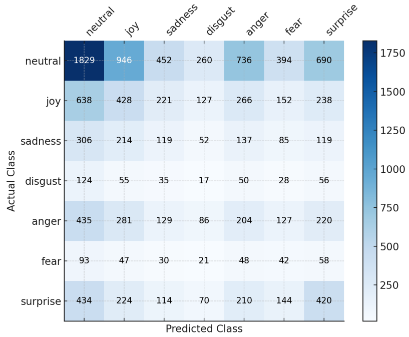

Figure 3: Confusion matrix for the Model 1

By looking at the Tables 2, 3, one can see that average Macro and Weighted metrics are higher in all given cases: F1-score, Precision, and Recall when dealing with GPT-3 Assistant with knowledge base. Metrics for emotions such as *sadness* for GPT-3 based Assistant without knowledge are relatively low compared to much better results in terms of metrics when dealing with GPT-3 based Assistant with knowledge base. Figures 1 and 2 present the confusion matrices for two version of GPT-3 classifier. For GPT-3 based Assistant with the knowledge base, more *neutral* cases were predicted correctly. In contrast, without the knowledge base, more *neutral* cases were predicted incorrectly as *joy* class.

4.1.2 BiLSTM model

The following Figures 3, 4 and Tables 4, 5 refer to the confusion matrix for Model 1 and Model 2 accordingly used in the BiLSTM experiment (please refer to Appendix A for training and model details).

While analyzing the confusion matrix and values of the metrics for the test data, one can see that the results are not the best - weighted accuracy is around 0.3, while recall and F1-scores are approximately 0.28. Results for both Models are pretty similar, so we can only say that context was not utilized by us well enough for it to affect prediction results (Figures 3 and 4).

The performance of the model was also affected by the distribution of the labels (refer to Figure 5) - such an unbalanced dataset caused labels for neutral, joy, surprise, anger (and also sadness) were more likely to be classified in the right way than

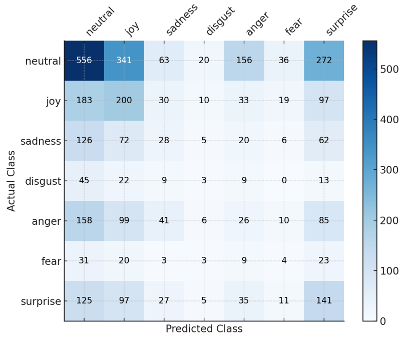

Figure 4: Confusion matrix for the Model 2

Emotion	Precision	Recall	F1-Score
neutral	0.47	0.34	0.40
joy	0.19	0.21	0.20
sadness	0.11	0.12	0.11
disgust	0.03	0.05	0.03
anger	0.12	0.14	0.13
fear	0.04	0.12	0.06
surprise	0.23	0.26	0.25
Accuracy			0.25
Macro Avg	0.17	0.18	0.17
Weighted Avg	0.30	0.25	0.27

Table 5: Classification Report for the Model 2

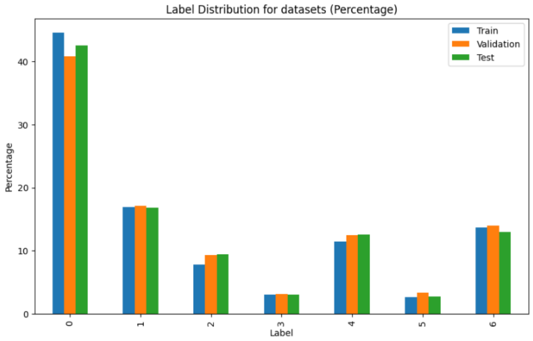

Figure 5: Label distribution in the dataset

fear and disgust ones.

When considering the best GPT-3 model with the knowledge base and all BiLSTM models, the GPT-3 overperforms the BiLSTM model in all presented metrics. It became clear that if we want to use BiLSTM models for the classification tasks where context plays an important role, there should be more complex preprocessing techniques and feature extraction for both the model input and the attention layer (both utterances and context values) that would be solved by utilizing GPT-3 model.

4.2 Subtask 1.2

Regarding fine-tuning the SpanBERT model, training and validation loss were calculated on the given dataset.

Two metrics were chosen to test the SpanBERT model. First is defined as **EM** or **exact match** and is defined as a sum of all of the individual exact match scores in the set, divided by the total number of predictions in the set. Also, the F1-score was used.

Parameter	**Value**
Learning rate	1e-5
Batch size	8
Training epochs	4
Weight decay	0.01

Table 6: Training config for subtask 1.2

Metric	**Value**
EM	21.42
F1-score	33.87

Table 7: Exact match and F1-score for Q&A task

Table 6 summarizes the training configuration. The parameters were set so that the **learning rate** is minimized, **batch size** does not exceed the given RAM of the machine, **training epochs** was set to between 2 and 4 according to BERT's authors' (Devlin et al., 2019) and **weight decay** was set to default.

Figure 6 presents the training and validation loss. The scores obtained for the Q&A task on the test dataset are shown in Table 7.

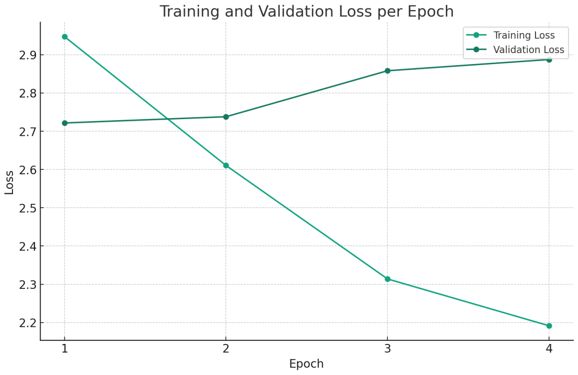

Figure 6: Training and validation in the epochs

4.3 Results Analysis

4.3.1 Models performance

In case of subtask 1.1, the acquired metrics for both assistant models demonstrate considerable promise. The model employed in this scenario was not pretrained, relying solely on its foundational capabilities as a Large Language Model (LLM). As anticipated, the model utilizing a knowledge base yielded superior results. We achieved a weighted F1-score of 0.52, accompanied by recall and precision values of 0.52 and 0.55, respectively.

For subtask 1.2, obtained metrics presented in Table 7 are much lower than metrics obtained in the SpanBERT case, where results on the SQuAD 1.1 were EM: 85.49, F1: 91.98. Given the nature of such models, metrics on our dataset should be close to the baseline set by SpanBERT.

4.3.2 Limitations and future work

As for subtask 1.1, textual data makes determining expressed emotion challenging due to the absence of non-verbal cues. With abundant data and resources for fine-tuning, models can predict emotions more efficiently. Despite precise instructions, models may occasionally "hallucinate" and provide unsuitable answers, interpreting emotions differently from the defined set of six instructions.

Much more attention should be paid to preprocessing and analyzing the train, val, and test dataset in subtask 1.2 to provide more meaningful and balanced questions and answers in a given context. The provided sizes of all datasets could be much higher to utilize fine-tuning training fully. The training parameters should also be applied more carefully, and hyperparameter tuning should also be used.

5 Evaluation and Conclusions

For the evaluation phase, we have used the evaluation data provided by the Organizers. The data was emotion-classified using GPT-3, and the data was suited for span extraction as in Section 3.2. We have also tried to use BiLSTM in this case, however, its capabilities were very limited when processing data with unknown words and short sentences (the probability of each occurrence of emotion was nearly identical). The results from SpanBERT were answers to questions built upon classified emotions. Obtained answers were added to the original evaluation dataset's utterances (called *main utterances* in the following text) based on the created by Author unique ID. Answers also contained spans of text that could occur in different utterances, so the utterances that did not belong to the main utterance were placed in different lists (meaning multiple cause spans) in each matched main utterance. Based on the main utterance answer and additional answers, emotion-cause pairs were created in a manner that the "emotion-cause_pairs" list contained emotion utterance along with its emotion category and a cause utterance ID followed by position indexes of predicted cause span within the utterance. The position index starts from 0, and the ending index is the index of the last token plus 1 excluding the punctuation token at the beginning and end. The evaluation phase ended for 1 entry uploaded on the CodaLab (Pavao et al., 2023) submission as follows: **w-avg. Strict F1**: 0.0449, **w-avg. Proportional F1**: 0.0723, **Strict F1**: 0.0462, **Proportional F1**: 0.0717, resulting in 9th place out of 29 teams. The results showed that each of the presented subtasks, namely emotion classification and emotion-cause pair extraction, could perform better in terms of classifying emotions and extracting spans. The changes could improve the overall score by employing GPT-4 architecture and experimenting with span extraction model architecture as well.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Shunjie Chen, Xiaochuan Shi, Jingye Li, Shengqiong Wu, Hao Fei, Fei Li, and Donghong Ji. 2022. Joint alignment of multi-task feature and label spaces for emotion cause pair extraction.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020a. ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3161–3170, Online. Association for Computational Linguistics.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020b. End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3574–3583, Online. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

A BiLSTM for emotion recognition in conversations

For the emotion classification task, the BiLSTM model was tested (Schuster and Paliwal, 1997).

A.1 Preprocessing

Preprocessing for this task was separated into a few different steps:

1. Duplicates elimination, as we have discovered that sometimes data might have duplicates within;
2. Special signs and stopword removal - in this case, punctuation and digits were removed from the text, then data was converted to lowercase, split into a list of words, and cleaned from English stopwords obtained from the NLTK library;
3. Text tokenization, indexing, and text to-sequence conversion - vectorization was done on the text by turning text into a vector based on TF-IDF and by fitting it to the processed text;
4. Sequence padding, to make sure that all of the input sequences are of the same length;

A.2 Model

The training dataset, presented by the SemEval competition's creators, contained information about a conversation between some group of people and emotion-cause pairs extracted from that conversation. The conversation consisted of multiple utterances, each with defined text, speaker, and emotion expressed by the speaker and their id.

For this task, such dataset was partitioned with a ratio of 0.6-0.15-0.25 to create, train, validate, and test datasets. Such a dataset was transformed into a set of objects, where each represents a single utterance along with information about the context (concatenated utterances within the conversation) and expressed emotion.

Two configurations were checked to establish which parameters would give the best result. All of the configurations used categorical cross-entropy (Lin et al., 2017) as a loss function, Adam (Kingma and Ba, 2014) as an optimization algorithm, and f1_score, accuracy, and recall were noted during all of the training stages. The model in each configuration had seven outputs, each for every primary emotion (Ekman, 1992), including neutral.

Layer (type)	Output Shape
utterance_input (InputLayer)	[(None, 250)]
context_input (InputLayer)	[(None, 250)]
embedding_12 (Embedding)	(None, 250, 250)
embedding_13 (Embedding)	(None, 250, 250)
concatenate_6 (Concatenate)	(None, 250, 500)
bidirectional_1 (Bidirectional)	(None, 250, 150)
attention_1 (Attention)	(None, 250, 150)
concatenate_7 (Concatenate)	(None, 250, 300)
global_max_pooling1d_1 (GlobalMaxPooling1D)	(None, 300)
dense_5 (Dense)	(None, 64)
dropout_2 (Dropout)	(None, 64)
dense_6 (Dense)	(None, 32)
dropout_3 (Dropout)	(None, 32)
dense_7 (Dense)	(None, 7)

Table 8: Second BiLSTM Model with Attention layer configuration

The first configuration (similar, but less complex than presented in Table 8) was a BiLSTM with two hidden layers and ReLU set an activation function, with an attention layer (for utterance data and without context) set with softmax as an activation function.

The second configuration shown in Table 8 was a BiLSTM with two hidden layers and ReLU set an activation function, with an attention layer for contextual data set with softmax as an activation function and an additional layer for 1D convolution operation.

A.3 Evaluation

We have trained both models using train and validation datasets and then tested them using the corresponding set.

A.3.1 Metrics

During the testing phase, loss function and accuracy were calculated for training data, and for the validation data were also calculated recall and f1 score. A confusion matrix was created for the test data.

A.3.2 Training and testing

Both models show the same tendencies for the training data, with the loss function decreasing with each epoch and accuracy getting better (Figure 7).

However, looking at the accuracy, f1 score, and recall, their values are pretty similar for the data in the same batch (Model 1 or Model 2); results for Model 2 are significantly better (Figure 8).

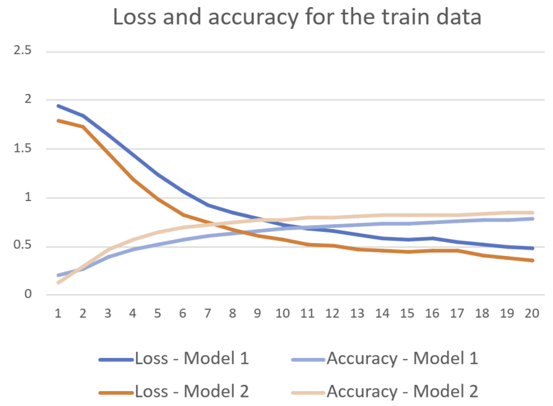

Figure 7: Metrics for test dataset in relation to the number of the epoch

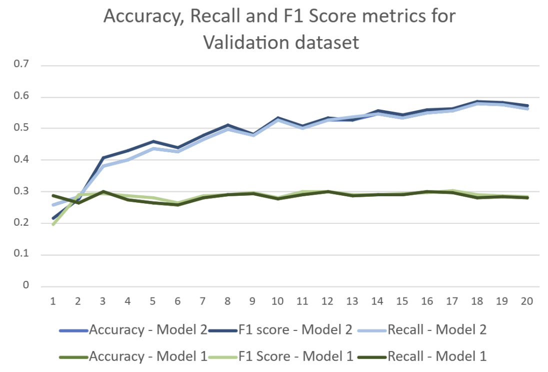

Figure 8: Metrics for validation dataset in relation to the number of the epoch