EDiReF subtask-III a wide model comparison
NLP course project

Antonio Lopez, Alessandra Blasioli and **Matteo Vannucchi**
Master's Degree in Artificial Intelligence, University of Bologna
{ antonio.lopez2, alessandra.blasioli, matteo.vannucchi }@studio.unibo.it

Abstract

This study explores emotion recognition intricacies in dialogues, addressing the SemEval2024 challenge with a dual focus on ERC and EFR. We center our experiments on BERT-based architectures. Starting with a simple baseline with no context we incorporated layers for capturing the global context, like an LSTM or an attention mechanism. This addition proves a relevant improvement over the EFR, but not on the ERC. To have a wider comparison we have also adapted EmoBERTa, well-known on the ERC task, to take into account the emotion-flips problem. From our findings, the attention mechanism ensures the best performance on the EFR while all the EmoBERTa architectures are significantly better on the emotion recognition task.

1 Introduction

Emotion recognition poses a formidable challenge due to its inherent complexity. One of the primary complexities stems from the contextual nature of emotions, where the same words can carry divergent meanings and evoke varied emotional responses based on the surrounding context. Studying the dynamics of emotional changes within conversations becomes even more intricate. Emotions are not solely contextualized within individual conversations, but are also influenced by the broader context surrounding them. Specifically common utterances like "What?" can have a different meaning depending on the context.

Our project sets out with the goal of tackling the realm of emotion recognition within the framework of the SemEval2024 challenge. The EDiReF challenge proposed at SemEval 2024 consists of two main tasks:

- **ERC**: which aims to classify the emotion of each utterance in a dialogue, where the emotions are taken from a predefined set.
- **EFR**: which aims to identify the trigger utterance (s) for an emotion-flip in a multi-party conversation dialogue.

The problem of emotion recognition has been addressed for a long time, but with the recent development in NLP and in particular with the introduction of BERT-based architectures (Devlin et al., 2019) several improvements have been made. Some of the most interesting are: (Gou et al., 2023) where BERT has been combined with an LSTM and (Huang et al., 2019) where BERT is pre-trained and fine-tuned on a specific dialog-based dataset. Of particular interest is EmoBERTa (Kim and Vossen, 2021) where a pre-trained RoBERTa model has been fine-tuned to have in input a full dialog divided into past, current, and future utterances. However, these methods only tackle the ERC problem. For the EFR task, we have (Kumar et al., 2023) that has used transformer encoders and stacked GRUs to capture the dialogue context.

In this report, we introduce a wide comparison between different models inspired by the above methodologies. Specifically, we broke our pipeline into two main components the **feature extractor** and the **CLF**. The first one is responsible for extracting features from the utterances, the second is for aggregation and classification. Different combinations of these two elements have been trained and tested.

Our experiments were all run on Google Colab Free Tier with the T4 GPU. For each experiment, we run 5 different seeds for a robust estimation. As for the dataset we used only the training set provided by the challenge's creators, for this reason, the results are not comparable with the ones obtained by other participants.

The findings of this study emphasize the superior performance of the models employing EmoBERTa as an encoder for the ERC task and the models that incorporate the dialogue context for the EFR. The encoder plays a crucial role in the architecture, as

highlighted by the significant performance difference between models using EmoBERTa compared to those with BERT. The inclusion of a layer capable of grasping the global context of the dialogue like an LSTM layer or an attention mechanism proves to be relevant for the model's efficacy.

2 Background

2.1 MELD dataset

The dataset used in this work is the one provided by EDiReF's creators. Specifically, it is a modified version of the MELD dataset, called MELD-FR, augmented with ground-truth EFR labels. It is organized into five different columns:

- **Episode:** a unique identifier for each dialogue instance.
- **Speakers:** an ordered list of the names of the speakers of each utterance.
- **Utterances:** a collection of sentences that represents a conversation or a fragment of a conversation between one or more speakers.
- **Emotions:** indicating the emotional classification of the particular utterance with seven possible values. This is a target for this work and is part of the ERC task.
- **Triggers:** a binary classification where one indicate an utterance that acts as a trigger for an emotion-flip, and zero when it does not. This is the target for the EFR task.

2.2 EmoBERTa

EmoBERTa is a model built upon the pretraining of the RoBERTa model by (Liu et al., 2019), designed to tackle emotion classification tasks in a dialogue context. The choice of RoBERTa, among various BERT-like models, was driven by its relatively simple structure and although the pre-trained model was not trained on more than two segments, EmoBERTa demonstrates its generalizability to three segments per input sequence. These segments represent past utterances, the current utterance, and future utterances in a dialogue. Each utterance is preceded by the name of a speaker to inform the model about the speaker's identity. EmoBERTa was trained on the MELD and IEMOCAP dataset.

3 System description

The proposed solutions are diverse, but a common starting point is the BERT baseline, which serves as the foundation for the structure of the subsequent models, developed to enhance the solution and its performance. As additional baselines, a random and a majority classifier were also implemented.

3.1 Baseline

The random classifier is nothing more than a simplistic classifier that takes input data and generates random predictions as output. On the other hand, the majority classifier makes predictions by always predicting the majority class. In this specific case:

- The majority-predicted class for emotions is the *neutral* class.
- For triggers, the majority-predicted class is zero, indicating the absence of an emotion flip.

For our baseline model, we implemented a straightforward architecture consisting of two primary components: an encoder and a classification head for each task. The encoder processes each utterance independently to generate embeddings. Then, the classification head comprises two linear layers, each of them interleaved with a ReLU activation function and a dropout layer for regularization. For the trigger classification task, the output of the classification head is then passed onto a softmax function. Similarly, the same process occurs for the emotion task.

We experimented with two training approaches for the baseline model: one involving fixed encoder parameters (models marked with *freeze* in their names) and the other incorporating training of the encoder parameters as well (*unfreeze*). When using the frozen encoder, we introduced a caching mechanism to speed up the training process. This cache allows us to avoid the expensive computation of the embedding for each utterance and focus only on the very fast computation of the linear layers. One obvious problem with this implementation is that the model does not have a global vision of the context, the next models try to address this limitation.

3.2 LSTM model

The LSTM model, following the work done by (Gou et al., 2023), is built upon the baseline and

integrates a BERT-based encoder with an LSTM layer and a residual connection. It can be seen in Figure 5. Throughout the training process, the output of the encoder is fed through the LSTM. This design allows the model to construct a comprehensive global context for each dialogue. The resultant global context is then concatenated with the output from the encoder (via the residual connection) and passed through the final classifier. The theoretical advantages of this model are:

- **Enhanced trigger prediction**: by using the global perspective of the entire dialogue, the model is expected to more accurately predict trigger utterances, since it should able to understand better the emotional progression throughout the dialogue.

- **Baseline-level emotion classification**: thanks to the residual connection, the model has access to the original sentence embeddings provided by the encoder. This should make the model obtain at least the same performance as the baseline in the emotion classification.

3.3 Attention Model

The Attention-based model closely mirrors the design of the LSTM model, but introduces a key variation: it replaces the LSTM layer with a multi-head self-attention mechanism, while still maintaining the residual connection. In this model the encoder's output is fed into the self-attention, this allows the model to build the global context of the dialogue, similarly to the LSTM layer. Differently from the LSTM, which builds the global context independently from a specific utterance, the attention mechanism allows the model to weigh and integrate different parts of the dialogue more precisely for the specific utterances. The performance should be similar to the LSTM model's.

3.4 EmoBERTa based models

These models are similar to the others described above, but with one major difference: the encoder. In this case, instead of using BERT we will be using EmoBERTa, provided by HuggingFace (Kim and Vossen, 2021). For these models the encoder is always frozen. Specifically, we implemented an EmoBERTa version of the baseline, one using LSTM, and one with attention mechanism, respectively called *emoberta-current*, *emoberta-lstm*, and *emoberta-attention*.

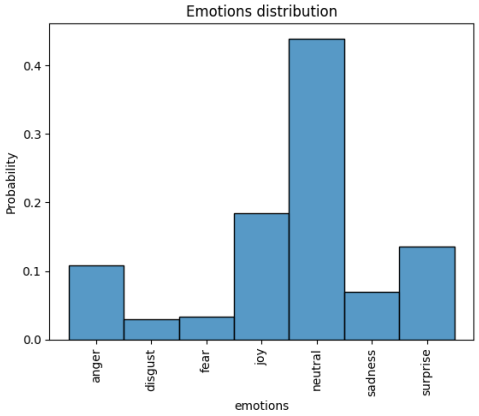

Figure 1: The label distribution for the emotions

3.5 Mixed model

This model, as shown in Figure 6, aims to leverage what we have identified as the strengths of the two encoder models, a point that will be demonstrated in the results section. Specifically, the model employs two encoders: one for the context, which is the concatenation of the entire dialogue, utilizing *bert-base-uncased*, a choice we determined to be optimal for the trigger classification task; and the second encoder, for the utterances, is chosen to be *emoberta-base*, optimal for emotion classification.

The use of two separate encoders allows the model to capture distinct representations for both the context and utterance, thereby capturing more detailed and specialized information for the task of emotion and trigger classification. The concatenation of these representations is then utilized as input for the final classifiers, enabling the model to make more precise predictions based on these composite representations.

4 Data

Due to the goal set for the task, we utilized only the training dataset among those provided by the challenge's creators. We chose to split the data into an 80% training set, 10% validation set and 10% test set.

The emotions are categorized into seven classes, encoded with values between 0 and n_classes − 1. The triggers are already distinguished by the values:

0 if there is no change in emotions.

1 if there is an emotion flip.

In figure 1 and 2 we can see how the labels are distributed. The classes are not very balanced and,

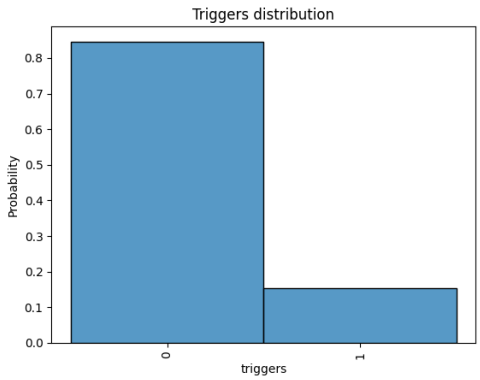

Figure 2: The label distribution for the triggers

for this reason, during training we will employ class weights.

Since different dialogues can have different lengths, we decided to pad dialogues in the same batch. We employed the following padding strategy:

- For sentences, we pad with an empty string. The value of the padding string is not critical, as it will not be encoded during the encoding phase. Instead, it's directly converted into a tensor of zeros, which helps to speed up computation.
- For triggers, the padding value is set to 2. Similar to the sentence padding, this value is not significant in itself. It is primarily used for structural consistency and is subsequently ignored in the loss and metrics calculations.
- For emotion, we pad with a value of 7. This follows the same rationale as for triggers.

We have devised two distinct strategies for structuring batches within the context of this project. Which strategy to use depends on the model being trained. The two approaches are:

- The string passed to the encoder is just the single target utterance. This strategy is the default one for every model.
- Alongside the target utterances a context is passed. This is just the entire dialogue concatenated together with information about the speaker of each utterance. This strategy is used only with the special *mixed-model*.

5 Experimental setup and results

All models were trained with the following common hyperparameters:

- **Learning rate**: $1 \cdot 10^{-3}$ for frozen models and $1 \cdot 10^{-5}$ for unfrozen ones. For the *mixed-model* we used $1 \cdot 10^{-4}$.
- **Batch size**: set as 32 for the frozen models and 8 for the others. An exception is the *mixed-model* that uses a batch size equal to 1.
- **Bert model**: for the BERT-based model we used *bert-base-uncased* as the encoder model, while for the EmoBERTa-based model we used *emoberta-base*. For the model named with *freeze* the encoder was frozen, while for *unfreeze* the encoder was part of the training.
- **Class weights**: which where calculated as:

$$w_y = \frac{n\_samples}{n\_classes * n_y}$$

  where n_y is the number of occurrences of samples of class y.
- **Hidden layers**: for the classification head there are two linear layers of size 128.
- **Dropout**: for the layer in the classification head is set to 0.2.

The learning rate and the number of hidden layers were first optimized using PyTorch Lightning's Tuner module. Despite exhaustive optimization efforts, the resulting values failed to yield any significant improvement in performance compared to the previously chosen values.

For the LSTM models, we used the following additional hyperparameters: the *hidden size* equal to 128, *number layers* to 2, and *bidirectional* set to False. For the attention models, we set the number of attention heads equal to 8.

For the training we used AdamW as the optimizer and `ReduceLROnPlateau` as the scheduler from Pytorch. For the loss, we used a masked cross entropy loss for both tasks, where the mask ignores the prediction associated with padding values.

We employed two primary metrics, namely two variants of the F1 Score:

- **The F1 score cumulative**, which is the standard Pytorch F1 score updated at each step and calculated only at the end of the epoch.
- **The F1 score dialogues**, which calculates the F1 score samplewise. The resulting F1 score is then the mean of each F1 score calculated for individual dialogues.

6 Discussion

The various proposed models demonstrate quite satisfactory results, particularly as illustrated in Table 1:

- For emotion, the best-performing model is the *mixed-model*, achieving a significantly higher score compared to the others.
- Regarding trigger, the *bert-lstm-freeze* outperforms the others in 'cumulative trigger' while *bert-attention-unfreeze* performs the best in 'cumulative trigger multiclass' and additionally shows superior performance in emotion classification tasks compared to *bert-lstm-freeze*.

From the results, it is evident that there is not a clear winner that outperforms the others across all the computed metrics. Certain models perform better on the triggers and others on the emotions, even more if we consider the dialogues metrics. Observing the results, it appears that EmoBERTa may be overfitting on this dataset, given the suspiciously high results obtained for the emotion task. This may happen because, as mentioned earlier, EmoBERTa is trained on the MELD dataset. In addition, the incorporation of a layer capable of capturing the entire context seems to be crucial, particularly in the task of trigger detection.

Comparing the obtained results with those produced by the baselines, we observe for the *mixed-model* that the emotion results have a higher value of at least 20 percentage point in F1 score, especially compared to the frozen baseline. However, for triggers, the values are very close to those of the baselines. Meanwhile, the *bert-attention-unfreeze* shows substantial increase trigger performance. Notably, dummy classifiers also exhibit satisfactory performance in trigger detection (if we consider the multi-class metric), achieving results close to the average of more advanced models. Nevertheless, there is a clear need for better and more accurate emotion predictions.

Finally, considering all the models and all the seeds experimented with, given the focus on the detection of emotion triggers in EDiReF 2024, we selected *bert-attention-unfreeze* as the model for the error analysis. The *bert-attention-unfreeze* model stands out as the superior performer, boasting an increase of six percentage points in 'cumulative trigger' F1 score compared to the baseline model. Despite this improvement, upon analyzing the confusion matrix for triggers, it becomes apparent that the model misclassifies 0 as 1 on 1221 occasions. This highlights that the primary challenge in the task remains in recognizing triggers, despite the model being the best performer in this aspect.

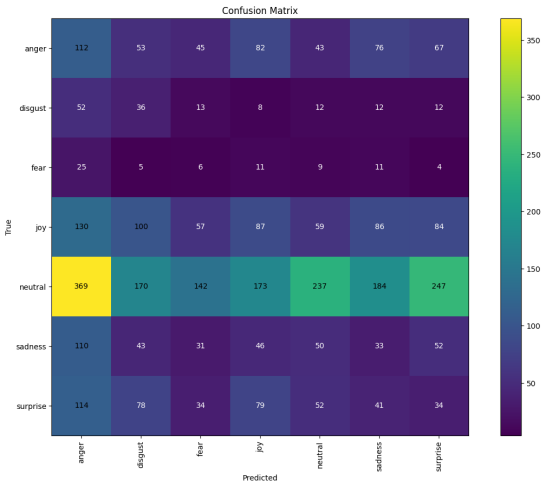

Figure 3: Confusion matrix for emotions

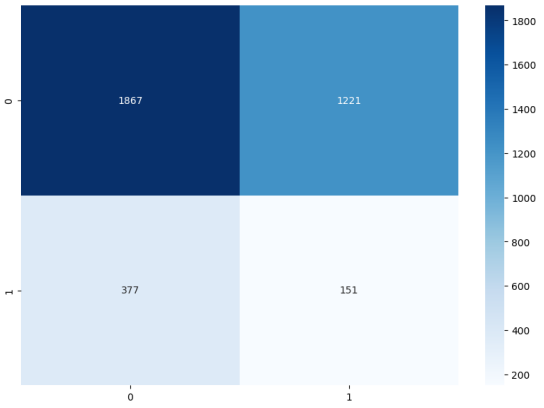

Figure 4: Confusion matrix for triggers

Regarding emotion, from the confusion matrix, we observe that the highest number of prediction errors occurs in the following cases:

- Neutral, confused with anger with 369 errors;
- Joy, confused with anger with 130 errors;
- Surprise, confused with anger with 114 errors;
- Sadness, confused with anger with 110 errors.

Overall, the most frequently confused emotion is neutral. This could be attributed not only to the inherent difficulty in distinguishing a neutral emotion, but also to the high prevalence of the neutral class in the dataset, which may lead to a higher number

of errors. It is noteworthy that *neutral* emerges as the majority class in the majority classifier.

Observing our selected model, the most challenging dialogues for emotions are those where the emotions involved are only *joy, neutral*, and *surprise*. This is likely because, as mentioned earlier, emotions like *neutral* may inherently be more challenging to distinguish due to their subtlety. Additionally, *joy* and *surprise* might share certain linguistic expressions, posing a difficulty in accurate differentiation. Additional analysis can be found in the Python notebook of the project.

7 Conclusion

In this study, we addressed the problem of emotions and triggers recognition within dialogues involving one or more participants. We implemented various models by combining BERT or EmoBERTa with mechanisms such as LSTMs or attention, experimenting with different architectures to find the optimal combination of elements starting from a simple baseline. The obtained results highlighted how there is not an outstanding model with respect to the others, depending on which metrics we give more importance a specific model can be preferred, ultimately, to balance project requirements and particularly meet the demands of trigger recognition, we chose to designate *bert-attention-unfreeze* as the best model. Despite this model's overall performance in emotion classification aligning with the average results of other models, it demonstrated remarkable values specifically concerning triggers. Future developments may involve enriching the dataset, firstly by using the full data provided by the challenge's creators. The use of context in emotion analysis has proven to be particularly effective; therefore, a further study on how to take advantage of it, for example following what is done in (Kumar et al., 2023), will be interesting.

8 Links to external resources

- **Dataset**: the dataset can be found here on the official page of the challenge.
- **GitHub repository**: link.

References

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

Zhinan Gou, Yan Li, and Xin Ning. 2023. Integrating bert embeddings and bilstm for emotion analysis of dialogue. *Intell. Neuroscience*, 2023.

Yen-Hao Huang, Ssu-Rui Lee, Mau-Yun Ma, Yi-Hsin Chen, Ya-Wen Yu, and Yi-Shin Chen. 2019. Emotionx-idea: Emotion bert – an affectional model for conversation.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, pages 1–10.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Table 1: Metrics for different models on validation and test set

Validation Set	**Cumulative emotion**		**Cumulative trigger**		**Cumulative trigger multiclass**	
Model	Mean	Std	Mean	Std	Mean	Std
bert-baseline-freeze	0.351	0.028	0.383	0.007	0.440	0.028
bert-baseline-unfreeze	0.405	0.010	0.392	0.006	0.512	0.011
bert-attention-freeze	0.343	0.027	0.392	0.004	0.535	0.073
bert-attention-unfreeze	0.417	0.013	0.398	0.008	**0.587**	0.006
bert-lstm-freeze	0.364	0.016	**0.402**	0.016	0.582	0.017
bert-lstm-unfreeze	0.405	0.013	0.394	0.007	0.504	0.021
mixed-model	**0.571**	0.015	0.370	0.009	0.500	0.026
emoberta-current	0.519	0.017	0.368	0.010	0.442	0.037
emoberta-lstm	0.489	0.018	0.392	0.015	0.536	0.046
emoberta-attention	0.512	0.011	0.364	0.006	0.451	0.083
random-classifier	0.121	-	0.283	-	0.448	-
majority-classifier	0.085	-	0.000	-	0.441	-
Model	**Dialogues emotion**		**Dialogues trigger**		**Dialogues trigger multiclass**	
	Mean	Std	Mean	Std	Mean	Std
bert-baseline-freeze	0.315	0.040	**0.394**	0.018	0.426	0.021
bert-baseline-unfreeze	0.366	0.012	0.370	0.015	**0.485**	0.009
bert-attention-freeze	0.317	0.043	0.348	0.039	0.432	0.035
bert-attention-unfreeze	0.379	0.013	0.334	0.019	0.478	0.009
bert-lstm-freeze	0.328	0.021	0.351	0.041	0.456	0.009
bert-lstm-unfreeze	0.372	0.017	0.383	0.010	0.477	0.018
mixed-model	**0.540**	0.013	0.357	0.020	0.476	0.021
emoberta-current	0.465	0.031	0.369	0.017	0.422	0.030
emoberta-lstm	0.436	0.025	0.350	0.054	0.424	0.037
emoberta-attention	0.467	0.013	0.352	0.033	0.384	0.075
random-classifier	0.073	-	0.264	-	0.409	-
majority-classifier	0.080	-	0.000	-	0.416	-

Test Set	**Cumulative emotion**		**Cumulative trigger**		**Cumulative trigger multiclass**	
Model	Mean	Std	Mean	Std	Mean	Std
bert-baseline-freeze	0.360	0.021	0.280	0.002	0.395	0.041
bert-baseline-unfreeze	0.408	0.013	0.288	0.008	0.472	0.015
bert-attention-freeze	0.333	0.022	0.318	0.025	0.507	0.094
bert-attention-unfreeze	0.407	0.024	**0.340**	0.008	**0.570**	0.013
bert-lstm-freeze	0.366	0.014	**0.340**	0.005	**0.570**	0.020
bert-lstm-unfreeze	0.414	0.010	0.293	0.008	0.461	0.020
mixed-model	**0.637**	0.010	0.275	0.006	0.466	0.037
emoberta-current	0.567	0.015	0.276	0.013	0.425	0.032
emoberta-lstm	0.608	0.020	0.314	0.013	0.540	0.057
emoberta-attention	0.584	0.011	0.274	0.014	0.428	0.085
random classifier	0.127	-	0.214	-	0.418	-
majority classifier	0.084	-	0.000	-	0.406	-
Model	**Dialogues emotion**		**Dialogues trigger**		**Dialogues trigger multiclass**	
	Mean	Std	Mean	Std	Mean	Std
bert-baseline-freeze	0.342	0.035	0.319	0.006	0.402	0.038
bert-baseline-unfreeze	0.400	0.017	0.313	0.011	**0.475**	0.016
bert-attention-freeze	0.325	0.048	0.281	0.029	0.407	0.054
bert-attention-unfreeze	0.400	0.016	0.294	0.011	0.462	0.017
bert-lstm-freeze	0.355	0.024	0.281	0.027	0.437	0.010
bert-lstm-unfreeze	0.404	0.027	**0.325**	0.010	0.458	0.023
mixed-model	**0.621**	0.007	0.292	0.023	0.464	0.028
emoberta-current	0.538	0.020	0.307	0.025	0.431	0.027
emoberta-lstm	0.564	0.019	0.257	0.037	0.416	0.044
emoberta-attention	0.536	0.011	0.279	0.037	0.380	0.078
random classifier	0.075	-	0.225	-	0.406	-
majority classifier	0.079	-	0.000	-	0.443	-

A Models architecture

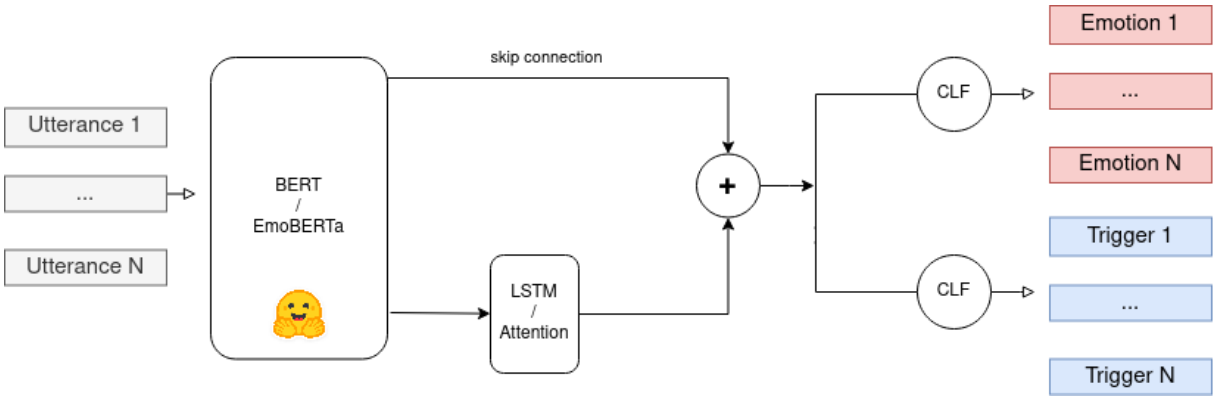

Figure 5: Basic architecture for all models that present an LSTM or an attention layer. For the baselines and for *emoberta-current* there is just a direct connection between the BERT based encoder and the CLFs

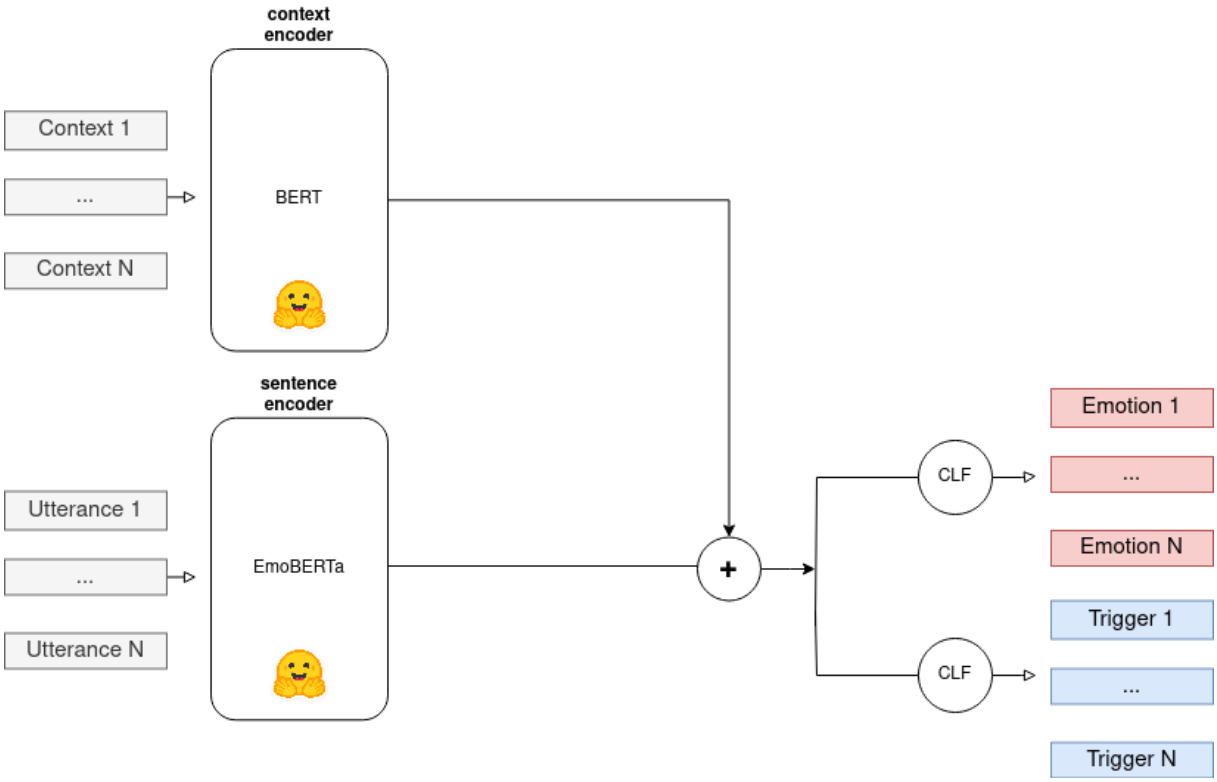

Figure 6: Mixed-model architecture, we can see how we employed two different encoders. The context encoder takes the concatenation of the dialogue and creates a representation of it. The sentence encoder instead work utterance per utterance. It is important to mention that in the context encoder the dialogue is divided into three segments: past utterances, current utterance and future utterances. For this the encoding of the context changes also if two utterances are in the same dialog