Emotion Discovery and Reasoning its Flip in Conversation
NLP Course Project

Alessandro Pasi, Razvan Ciprian Stricescu and **Matteo Belletti**

Master's Degree in Artificial Intelligence, University of Bologna
{ alessandro.pasi8, razvancipr.stricescu, matteo.belletti5 }@studio.unibo.it

Abstract

Our objective is emotion detection and emotion-flip identification in multi-party conversations by exploring various model architectures and techniques. We implement a comprehensive approach involving the creation of multiple models, including BERT-based architectures inspired by recent research papers. We preprocess the data, split it at the dialogue level, and evaluate the models on different evaluation metrics. Our methodology also includes training with different seeds to assess model robustness. We created baseline models for comparison purposes and then experimented with Bert-based models both frozen and unfrozen, with the latter consistently yielding superior results. Surprisingly, we observe an inverse relationship between the amount of context provided as input and model f1-scores. These insights provide valuable implications and discussion topics for the importance of context in this peculiar task.

1 Introduction

The problem we are addressing involves identifying the trigger utterance(s) that lead to an emotion-flip in multi-party conversation dialogues in English, while classifying each sentence into one of seven given emotions. An emotion-flip refers to a significant shift or change in emotional expression during a conversation. Solving these problems could be important for many reasons among which: It enhances natural language understanding in conversational AI systems, crucial for applications like customer service, chatbots and virtual assistants. Improved human-machine interaction is achieved by recognizing triggers and emotions, fostering emotionally intelligent conversational agents. The complexity of emotion recognition in multi-party conversations contributes to understanding intricate social dynamics, applicable in team collaboration, group decision-making, and online discussions. Lastly, applications in mental health and

well-being benefit from identifying emotional triggers, particularly in therapeutic chatbots, enabling more personalized and effective interventions.

Various techniques can be employed for this problem, including supervised machine learning, deep learning models, context-aware models, rule-based systems, ensemble models, transfer learning, hybrid approaches and online learning. These approaches offer specific advantages, such as capturing complex patterns or considering contextual information, but also pose limitations, including the need for large labeled datasets or potential challenges in model integration. The choice of approach depends on factors such as problem characteristics, available data, and computational resources, often requiring a combination of methods or a carefully tailored model for effective identification of triggers in complex conversational dynamics.

Our approach involves the creation of multiple models and the evaluation of various techniques to discern which methods contribute most significantly to result improvement. By developing a diverse set of models employing distinct methodologies, we aim to comprehensively assess their performance and identify the most effective strategies. This approach not only allows us to optimize results but also enhances our understanding of when and how specific methods excel in this specific task.

To set up the experiment we proceeded as follows: First, we tackled the issue of NaN trigger labels by converting them to zero, ensuring proper formatting and preventing errors during processing. We then split the dataset into 80/10/10 train/validation/test sets. Moreover, we created two baseline classifiers, specifically a random and a majority classifier, to have a starting point for comparison with our personalized Bert models, which we trained both with frozen and unfrozen embedding layer. Finally, we evaluated and compared the

results.

For model evaluation, we included a Sequence F1 and Unrolled Sequence F1 as metrics, computed for both emotions and trigger labels. To ensure robustness, the models are trained and evaluated on five different seeds. Finally we reported the average and standard deviation on the labels, providing a comprehensive understanding of model performance.

2 Background

We experimented with different techniques inspired by recent advancements in emotion recognition within conversational contexts, with various degrees of success. The one we mainly used and learnt from draws inspiration from the paper (Yang and Shen, 2021) which addresses the task of emotion dynamics modeling using BERT. In the paper, traditional approaches relying on Recurrent Neural Networks (RNNs) are replaced with BERT-based models, including Flat-structured BERT (FBERT) and Hierarchically-structured BERT (H-BERT), designed to capture inter-interlocutor and intra-interlocutor dependencies. The Spatial-Temporal-structured BERT (STBERT) further identifies emotional influences. Evaluation on benchmark datasets indicates a significant performance improvement, surpassing state-of-the-art baselines. Using this knowledge, we deduced that incorporating additional context could be crucial for achieving better scores.

3 System description

For the task at hand we first implemented two dummy models using DummyClassifier from the library *sklearn*, which we used with "uniform" and "most frequent" strategies. These dummy classifiers serve as baseline comparison for our personalized model which is based on BERT, more specifically on a pre-trained, uncased version of it as we think that casing wouldn't have a significant influence for this particular task.

The BERT models all share the same architecture, as seen in 1, which is composed of the following: BERT Encoding layer, Dropout layer and a Linear layer.

On top of the encoding layer we added two personalized classification heads for the distinct labels 'trigger' and 'emotion.' They are composed of two layers each: one dropout layer with dropout rate set to *0.3* and one linear layer with seven output
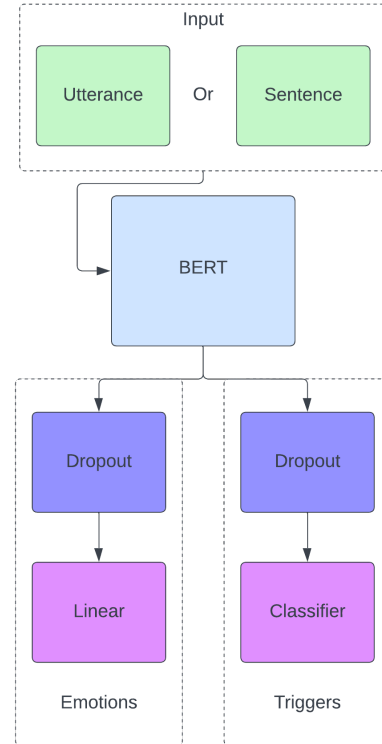

Figure 1: Simple diagram showcasing the BERT models pipeline.

channels for the emotions and two for the triggers. The two heads independently process the output generated by the encoding layer, learning different features for the two tasks we want to solve. This approach proved much more effective than the first attempt where we tried to use a single classification head. In this discarded, early attempt we had "fused" together trigger and emotions trying to predict them both at the same time. Laterm, we realized by analysing performances that the model had some trouble predicting both emotion and triggers in a joint fashion, leading us to implement the two-head version, which allows for more specialization and precision.

We also experimented with LSTMs and BiL-STMs as classification heads but they proved to be overly complicated as the results of the "simpler" models were still better.

4 Data

The dataset we used for the task at hand is composed of 4,000 dialogue samples taken from the TV show Friends. We divided the data into training, testing, and validation sets with an 80/10/10 split. The data is organized into 4 columns: 'speakers,' 'emotions,' 'utterances,' and 'triggers,' where an
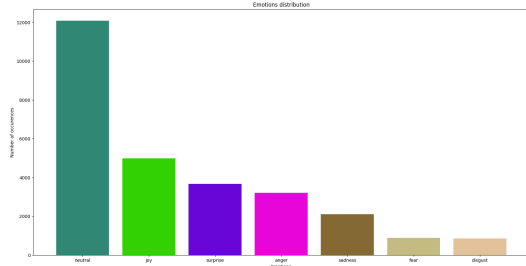
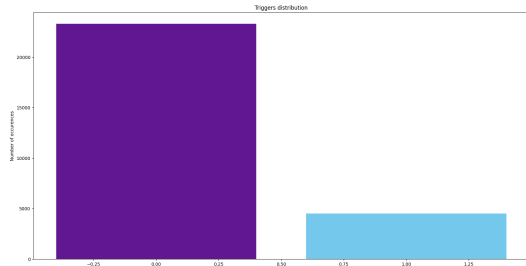Figure 2: Emotion distribution on the MELD dataset.

Figure 3: Trigger distribution on the MELD dataset.

utterance is said by a speaker and characterized by an emotion and a trigger. Emotion describes the feeling expressed in the utterance while trigger is a binary value set to 1 when the utterance produces an emotion flip in the dialogue. However, it appears to us that emotion flips (triggers) are not consistently or intuitively labeled with '1'. This made it challenging for us to understand the policy adopted for triggers. As both the macro-models don't require any particular pre-processing techniques to work we simply presented them as they are with just some minor adjustments, the only important differences being dependant on the strategy used.

Fist thing we replaced NaN values in the trigger columns with zeros to prevent errors. While visualising the data distribution, as shown in 2 and 3, for the target labels we observed heavy class imbalance: emotions toward the neutral label and triggers toward zeros. For this reason, we later adopted weight balancing to try and give more importance to the label inversely proportional to its percentage among the total data.

For the BERT models we first tried a teacher forcing technique by asking for a sentence-by-sentence label prediction on the whole context. To achieve this, for sentences from the second one onwards, we provided the previous sentences in the same dialogue and their correct labels. However, this approach didn't work quite well, and we think it might be due to the dialogues being so short that

context isn't of much help and ends up introducing unnecessary noise. Nevertheless, this approach deserves new attention for future work.

The next approach was way simpler; we provided the model with single sentences and asked for a single label and trigger prediction. This worked remarkably well, leading to a great improvement in the results compared to the previous data organization, which included the context. Even though this is intuitive for the emotion label, as we would think a sentence shows an emotion in a semi-independent fashion from the context, it's not as intuitive for the trigger, which we would think is more context-related.

5 Experimental setup and results

Our work and experiments were run both on Colab and locally on our machines' GPUs. All the architectures proposed and tested have in common the same BERT layer as encoder and were all initially run, for quick testing purposes, on a smaller version of it, proposed by (Bhargava et al., 2021).

The first model we tested used a single head classification layer, composed of simple dropout and linear layers, and used what we later started to call the "context" dataset. This dataset was born out of the necessity to classify one emotion and trigger at a time as we only possessed a single classification layer. Instead of giving the whole utterance as input to the model we repeated it as many times as the number of sentences, also concatenating the triggers and emotions not being predicted as to do "teacher forcing" passing the model the whole dialogue context each time. Although promising, this model didn't perform as well as we thought, having something to do with how we classified emotions and triggers. As we used only one classification layer we thought of fusing together emotions and triggers, thus actively doubling the amount of classes we wanted to predict. This solution proved problematic as it couldn't classify many of the minority classes such as disgust and trigger 1 together. We tried different classification layers like LSTMs but still couldn't make it work.

Our second implementative approach, which is also the one we currently employ, deploys two similar classification heads both composed of a dropout layer followed by a linear layer, but now split in two branches in order to solve the problem of having to classify different macro-labels. Initially we employed the "context" dataset but after some trials

we found that the minority classes were still hardly classified. This new problem made us question the importance of context in this peculiar task so we devised a new dataset, a way simpler one, that divides each utterance in single sentences and feeds them separately to the model. To our surprise the model results improved so we decided to further expand and explore this solution.

After selecting an architecture for the unfrozen BERT model, we also implemented a frozen version that does not train the embedding layers. This was done to assess the performance impact of these layers. We later proceeded in searching for the correct hyper-parameters for them. The first thing we decided after some trials was to decrease the batch size, which we originally set to sixteen, to a single batch as a larger batch size didn't allow us to physically run the models on our machines. For the remainder of the hyper-parameters instead of trying a grid-search approach, which we feared would be too time-consuming as the model was already big enough, we decided upon using the ones proposed in the paper (Devlin et al., 2019), such as hidden output channels and learning rate. As a compromise we also chose two epochs to train upon in order to get good results and also save time between trials, as each epoch would take about forty-five minutes to complete.

As optimizer we chose Adam as it is both a popular one and has been shown to be effective in a variety of NLP tasks, including text classification, named entity recognition, and machine translation, while also requiring less tuning of hyper-parameters compared to others, something we were concerned about as explained before. Our initial scheduler of choice was a simple linear scheduler with warm up. We believed it was particularly useful in this task and similar ones as the warm up phase helps the model converge faster by allowing it to initially explore a larger region of the parameter space. In latter stages we decided to switch to another one which we think is even "simpler", *StepLR* from *torch_optim*, doing a step each epoch instead of each batch. We kept this change as we didn't observe any peculiar changes in model performance while potentially removing some time-consuming overhead.

In order to evaluate our models we focused mainly on f1-scores, to be precise macro and weighted, which we computed thanks to the functions already present in the *sklearn* library. The

first model we tested used as loss function either binary or not cross entropy with logits but we found that neither could really deal with the dual nature of the classification task, so we decided to use a custom loss in our second approach with the multi-headed classification. As emotion classification is a multi-label problem we use cross entropy while for triggers, which is a binary problem at its core, we use binary cross entropy. Our custom loss is just the sum of these two losses computed on their specific classification task, enabling our model to learn in a better way this dual nature. Although we used our custom loss our model initially still wasn't able to classify minority classes, at least not in the capacity we wanted, thus we added class weights to our loss functions to counter the class imbalance in the dataset. The class weights are computed using the *compute_class_weights* from *sklearn*, which is simply the number of samples of that class divided by the number of classes times the binary count of classes. This simple exploit resulted in our model classifying even the minority classes in a robust way.

We further evaluated our models based on two different aspects: unrolled sequences and dialogue-based. In the unrolled sequences aspect we simply analyze our model by considering all sentences regardless of utterance, thus prioritizing a overall performance. In the dialogue-based approach, called sequence, we analyze it based on each dialogue, thus accentuating the "context" nature of utterances. In 1 we present our results on both the unfrozen models with "context" and without on the test set.

6 Discussion

As we deployed two similar models but with totally different data strategies we were really interested in seeing how they would fare. Based on table 1 we can observe that the model without "context" has way better results, both in weighted and average f1-scores across all seeds, than the model that uses the whole utterance. This initial observation also sparkled in us a question about the importance of context in this task. In order to better analyze this behaviour we looked upon other metrics such as the unrolled and the sequence f1-scores previously mentioned. In table 2 we can see that for both metrics we can see that it's still in favour of the latter model.

Context is an important factor in natural language processing (NLP) tasks. Context analysis

Seed	4	42
	Majority baseline	
Macro F1	E: 0.08 T: 0.46	E: 0.08 T: 0.46
Weighted F1	E: 0.25 T: 0.79	E: 0.25 T: 0.79
	Random uniform baseline	
Macro F1	E: 0.13 T: 0.43	E: 0.13 T: 0.44
Weighted F1	E: 0.17 T: 0.57	E: 0.18 T: 0.58
	Model with "context"	
Macro F1	E: 0.15 T: 0.46	E: 0.17 T: 0.46
Weighted F1	E: 0.31 T: 0.79	E: 0.30 T: 0.79
	Model without "context"	
Macro F1	E: 0.20 T: 0.46	E: 0.32 T: 0.49
Weighted F1	E: 0.38 T: 0.79	E: 0.47 T: 0.78

Table 1: Results of the two best seeds out of five different ones. E stands for Emotion while T for Trigger.

Seed		4	42
	Model with "context"		
Unrolled	Macro	E: 0.15 T: 0.46	E: 0.17 T: 0.46
	Weighted	E: 0.31 T: 0.79	E: 0.30 T: 0.79
Sequence	Dialogue average	E: 0.16 T: 0.45	E: 0.14 T: 0.45
	Model without "context"		
Unrolled	Macro	E: 0.20 T: 0.46	E: 0.32 T: 0.49
	Weighted	E: 0.38 T: 0.79	E: 0.47 T: 0.78
Sequence	Dialogue average	E: 0.31 T: 0.45	E: 0.34 T: 0.46

Table 2: Results of unrolled and sequence metrics on the two best seeds out of five different ones.

in NLP involves breaking down sentences to extract the n-grams, noun phrases, themes, and facets present within. The goal of context determination is to find answers to four questions: Who is talking? What are they talking about? How do they feel? Why do they feel that way? All these questions are pertinent with the task at hand however the model without the context outperformed the one with it. We tried to calculate the correlation between the emotions, triggers and the context and we got as a result that only 1.6% of emotions and 35% of triggers are dependant on context, a result which may give further credibility to the simpler model. Even looking at table 3 we can observe that the latter model has better standard deviation than the other one.

Initially these results made us question the correctness of our model which employs "context", whether teacher forcing techniques or the model itself couldn't perform because of innate flaws with its design. While we still very much leave this possibility open, we focused our effort in trying to analyze the task and the dataset at hand in a smarter way.

Initial data analysis, as already reported in previous paragraphs, told us that data distribution is quite imbalanced, thus giving a partial explanation to why the models couldn't reliably classify minority classes. We can see from figure 4 that the model can classify the majority class, "neutral", fairly well while it struggles for the minority class, like "disgust" or "fear".

One of the ways we think this problem could be solved is with further class balancing by using either dataset expansion or better and more thorough class weighting techniques. We also believe that the "context" model could be improved as we are uncertain about our architecture design efficiency. Another important point we want to further research is context importance based on task. The "without context" model surpassed our initial estimates for the scores obtainable on this task and likewise our interest grew about whether a more complex model is really necessary.

Finally, as anticipated, we observed that the frozen BERT models were consistently outperformed by the model with the capability to learn the most suitable embeddings, especially the models "without context", thereby highlighting the importance of this process as seen in table **??**.

7 Conclusion

In conclusion, our study involved a total of five models, including three BERT models trained in both frozen and unfrozen configurations, resulting in a total of eight different model to evaluate (each with five different seeds). The obtained results generally aligned with our expectations. The

Seed	4		42	
RMSE	U	S	U	S
w "context"	E: 2.12 T: 0.38	E: 0.09 T: 0.04	E: 2.27 T: 0.38	E: 0.09 T: 0.04
w/o "context"	E: 1.99 T: 0.38	E: 0.16 T: 0.04	E: 1.94 T: 0.41	E: 0.16 T: 0.06

Table 3: Root mean square error on unrolled and sequence metrics on the two best seeds. U stands for Unrolled while S for Sequence.

Figure 4: Example of miss classifications on a dialogue.

Model	Emotion		Trigger	
Macro F1	Mean	S.d.	Mean	S.d.
Majority	0.8	0	0.46	0
Random	0.126	0.005	0.428	0.008
BERT context	0.11	0.064	0.46	0
BERT context frozen	0.08	0.066	0.46	0
BERT w/c	0.234	0.097	0.468	0.013
BERT w/c frozen	0.198	0.004	0.46	0

Table 4: Report of the mean and standard deviation of the Unrolled F1 score.

Model	Emotion		Trigger	
Macro F1	Mean	S.d.	Mean	S.d.
Majority	0.08	0	0.46	0
Random	0.126	0.005	0.428	0.008
BERT	0.124	0.036	0.45	0
BERT frozen	0.124	0.036	0.45	0
BERT w/c	0.3	0.069	0.454	0.005
BERT w/c frozen	0.318	0.008	0.452	0.004

Table 5: Report of the mean and standard deviation of the Sequenced F1 score.

majority classifier performed well only on the '0' trigger and neutral emotion categories, making it the least effective overall, while the random classifier exhibited an average performance as a simplistic model. More interesting consideration can be done for the BERT models, especially the unfrozen ones, which consistently outperformed their frozen counterparts, leveraging the training of embedding layers, although being computationally heavier. For these models, contrary to expectations, we observed that the accuracy improved inversely proportional to the amount of context provided as input. As a result, we concluded that for the task at hand, context is not as crucial as we initially expected. Treating sentences as more independent significantly improved the accuracy achieved. This phenomenon may stem from the limitation it imposes on overfitting, while at the same time facilitating a faster convergence of the loss function.

Promising avenues for future improvement include exploring advanced model architectures that can efficiently handle larger contextual information without compromising computational efficiency.

8 Links to external resources

References

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

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

Haiqin Yang and Jianping Shen. 2021. Emotion dynamics modeling via bert.