Meta-Learning in Conversational Context

Using deep learning models based on meta-learners to produce accurate emotion predictions in user generated conversations

Barbara Gendron-Audebert

Under the supervision of Gaël Guibon, Associate Professor at LORIA Laboratory (SyNaLP team), Nancy, France

A thesis presented for the degree of
Master in Mathematics

Faculty of Science, Technology and Medicine
University of Luxembourg
Belval Campus, Esch-sur-Alzette
June 2023

Acknowledgements

I first wish to express my sincere thanks to Gaël, my supervisor, whose dedication and ambition ensured me a constant progress up to this production, which would not have been as accomplished without his guidance. His insightful feedback helped me to develop elaborate technical reasoning, always in a positive and benevolent atmosphere.

I would also like to thank Christophe, SyNaLP team leader at LORIA laboratory, for welcoming me in the team, for his availability and involvement during my internship. I am really grateful towards all the SyNaLP members, who have made these few months one of the most valuable research experiences I have ever had.

Then, I owe much credit to the University of Luxembourg and the École Nationale Supérieure des Mines de Nancy for the quality of their teaching during my specialization in Data Science, which for sure allowed me to apply for such a position.

Finally, I want to express my heartfelt gratitude to my family for their unwavering love, support, and simply constant presence, which has been my greatest source of strength throughout this journey.

Contents

Introduction

In recent years, the field of Natural Language Processing (NLP) has witnessed remarkable advancements, revolutionizing the way humans interact with machines and transforming various industries. NLP is an interdisciplinary branch of artificial intelligence that focuses on narrowing the gap between human language and computers, enabling machines to understand, process, and generate human-like text. Artificial Intelligence (AI) approaches have been helping to improve both natural language understanding and generation. In addition, with the exponential growth of available data amount and the increasing reliance on AI-based assistants such as ChatGPT[1], NLP has become more necessary than ever before. Thus, this Master Thesis explores the continuous relevance of NLP in shaping the future of human-machine interaction through conversations analysis which has many applications to be developed later in this work. Unraveling the intricacies of conversation features can lead to unprecedented possibilities and implement robust technologies that truly understand and converse with humans.

Conversations are omnipresent in User-Generated Contents (UGC). They can be found in social networks, in fictional creations such as films and TV shows, or through human-machine interaction *via* chatbots. Then, Emotion Detection in Conversation (ERC) is an important part of conversation understanding, and many use-cases can be found to emotion-oriented conversation understanding. For instance, in healthcare, some specific chat bots have already been built in order to provide a daily follow-up of patients with psychological or psychiatric disorders [22]. This can be done by analysing sentences written by the patient [45] or answers to specific questions [6]. This technology makes it possible to monitor patients much more frequently, and to identify trends towards relapse or instability [70], for instance. Besides, in the business field, emotion detection can bring useful insights that help to enhance recommender systems [65] and customer services [26].

For all these purposes, the advent of deep learning has been a complete revolution. Neural networks based models can now provide competitive solutions to most of the NLP concerns, including conversation understanding and, in particular emotion detection [1]. A specific deep learning models family seems adapted to address such challenges: meta-learning algorithms [61]. It basically consists in learning to learn or learning to generalize from multiple tasks or datasets to acquire knowledge or prior experience that can be generalized to unseen tasks or domains. In our case, meta-learning can be used to effectively learn the relationships between emotions. Indeed, there is a broad plurality of emotions, which requires to focus on general concepts. In this work, that's exactly what we are trying to do by considering the conversational context, which would be a huge step forward in ERC.

The main goal is to use deep learning models based on meta learners to produce accurate emotion predictions in user generated conversations. At the best of our knowledge, the latest methods that use meta-learning to do ERC do not integrate the conversational context. The goal of this Master Thesis is therefore to adapt existing methods, not only metric-based meta-learning approaches, but also all kinds of meta-learning schemes.

[1] `https://openai.com/blog/chatgpt`

In the next chapters, we will gradually dive in meta-learning for emotion recognition. Starting with an overview of the key concepts in chapter 1, we will then explore in more details the deep learning tools involved, as well as the associated mathematical formalisms (Chapter 2). The next chapter will be devoted to examining some existing meta-learning approaches, which leads us to the search for a suitable method for integrating conversational context into a meta-learning framework (Chapter 3). Then, we will detail the elected experimental setup along with the results (Chapter 4). Especially, we found that tuning a meta-learning model that accounts for conversational context is a demanding task in terms of training, with a weighted F1 score of 51% in multiclass classification. Nevertheless, the relevance of the predictions is encouraging, suggesting a favorable basis for generalization. Finally, we conclude this study by pointing out the limitations and perspectives for future work suggested by our approach (Chapter 5).

1. Context

In this first chapter, we describe the essential concepts that constitute the basis of what we will study in the following chapters. We begin by describing the type of textual data to be processed, before describing the task performed on such data. We conclude with a brief description of the family of methods chosen for this study.

1.1 Conversations in NLP

From a technical point of view, conversations belong to the family of UGC. This gathers a collection of user-generated text, audio, or video content that is created and shared by individuals within a digital platform. Conversations, in particular, can be considered a form of UGC when they occur in online communication channels such as social media platforms, discussion forums, messaging apps, or chatbots. From such contents, there exists NLP techniques to process conversations, which implies to convert them to a computer-friendly format. First, the conversation is often stored as a list of *utterances*, where an utterance is defined as a whole speaker turn. Then, each utterance will be converted in a series of phrasal entities called *tokens* to end up with a meaningful input format for applying NLP methods. From this processed dialog, many tasks can be performed, such as speaker detection, topic prediction or emotion detection. It's the latter in particular that will be the focus of this work.

1.2 Emotion Detection

To perform such a task, using *machine learning* models seems to be a relevant choice as they can deduce patterns and features indicative of emotions for large amounts of data, and generalize that knowledge to accurately classify emotions in previously unseen conversations. Emotion detection therefore belongs to the family of classification tasks. Figure 1.1 shows a basic example of emotion classification on different sentences from a dialogue. In this project, we focus on dialogues with only two interlocutors, known as dyadic dialogues. Since emotions are predicted from utterances in conversations, this task is exactly ERC, whose advances have been reviewed by Poria et al. [53].

In this work, one of our goal is to broaden this typical classification scenario in order to include cases where we find other labels. These new emotions can be variations of the initial classes or more general feelings. One can also imagine a classification scenario on previously unseen classes, as it is done in recent work [25]. We use meta learning in this work to seek comparable performances for classification task in this much more flexible scenario. However, to understand meta-learning as such, it is necessary to first address what it is based on: deep learning.

1.3 Deep Learning

Contrary to regular machine learning where features are provided by the model designer, the *deep learning* field includes machine learning models able to automatically extract

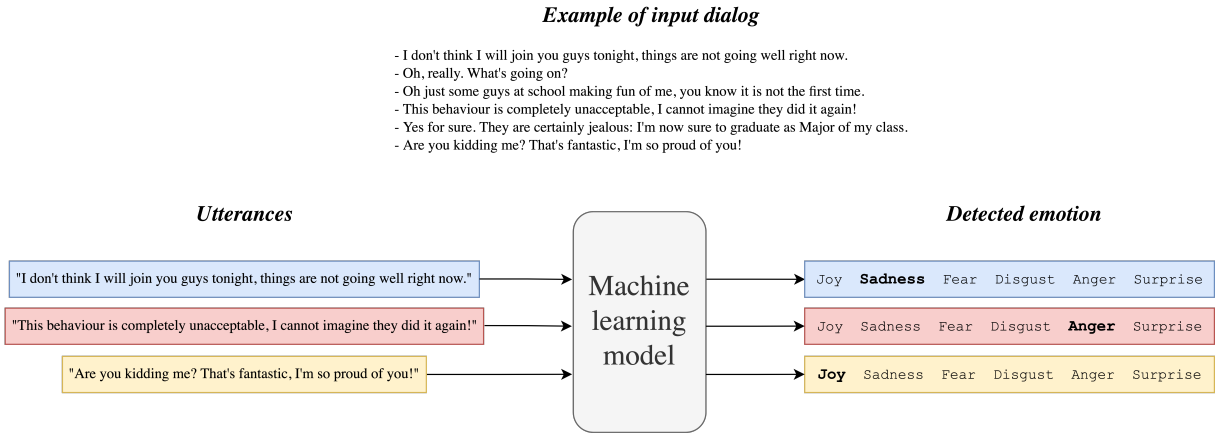


Figure 1.1: An example of emotion recognition on utterances from a dyadic dialog. The six emotions classes correspond to Ekman's six emotions.

supposedly relevant features. In deep learning, one uses *neural networks* to perform feature extraction. Neural networks training is basically what deep learning is all about. A neural network consists of interconnected nodes (therefore called neurons) organized in layers, where information flows through the network, allowing it to learn patterns, make predictions, and perform various tasks such as image recognition, text classification or clustering. The learning is actually performed through iterative parameters updates. Indeed, each neuron is assigned a value called a parameter, and parameters are then updated throughout the training phase, so that at the end of this stage the network acquires a certain understanding of the input data. It appears that deep learning has outperformed shallow machine learning models in many application (forecasting, image processing, ...). Therefore, the unequalled performance of neural approaches has made them an almost systematic choice in the development of NLP systems. One of the main advantages of using deep learning for NLP is flexibility in the model structure, especially the input layer, which enables sequential data (such as text) to be processed efficiently. Moreover, even though neural networks have been around since the 1980s [21], modern numerical methods make it possible to calculate backpropagation (*cf. infra*) in an approximate way, and this, combined with increasing machine capacities, makes it possible to stack many layers of neural networks and make computations on it. This is important because the depth of the network allows to describe data with more details, revealing some more subtle features.

Since deep learning methods seem adapted to emotion recognition, it is now time to dive deeper in the technical background required to describe both the experimental setup and state-of-the-art in application of meta-learning approaches on conversations.

2. Background

Now that we have covered some general ideas about the method, let's have a closer look at how we are going to leverage deep learning in our study. This chapter provides technical insights about key concepts in deep learning and how to train such models.

2.1 Deep Learning

Traditional architectures for sequential data. As text data consists of sequences of words, a usual way to deal with sequential data before the advent of deep learning was to use Conditional Random Fields (CRF) [33]. It is basically a Hidden Markov Model (HMM) [44] that aims at maximizing the set of probabilities of the sequence with a softmax. In the case of classification, the probabilities are for each class, and the aim of CRF at inference is to evaluate the probability of having each class given the sequence. A common way to perform this evaluation is to use the Viterbi algorithm [18] which aims at finding the most probable sequence of states given a sequence of labels

Now, considering deep learning methods, it is important to use a neural network design adapted to this sequential nature. That's exactly the aim of Recurrent Neural Networks (RNN) [60, 31]. In such structure, the nodes are connected to each other to form a chain, allowing information to be processed sequentially. Nevertheless, RNN have some issues, especially they struggle to account for long term dependencies because of the vanishing gradient phenomenon. This refers to the diminishing magnitude of gradients as they are back-propagated through many nodes of the RNN, leading to ineffective learning as most gradients are equal to zero.

A first improvement. A variant of RNN, named LSTM for Long Short-Term Memory Networks [27], has been developed to overcome the vanishing gradient issue. Like RNN, LSTM have a cell chain structure, which makes them suitable as well for sequential data processing. However, the inner structure of each cell is different, in the sense that information circulates along two different pathways that can be likened to short- and long-term memories. The latter solves the problem of long-term dependencies encountered with RNN. What is particularly interesting about this memory is that it retains the word's context, and LSTM can be used to process a sentence in both ways (bidirectional LSTM) to account for the whole surrounding context [51]. Even though, as the cell structure is more complex in LSTM, a model based on such architecture requires heavy computations that are not parallelizable due to its sequential nature. This can be limiting when stacking several layers, also stacking LSTM layers has been proven to be quite unstable.

The advent of Transformers. Furthermore, a new model architecture manages to solve this computational challenge: the *Transformer* [68]. In addition to being quite optimal thanks to parallelizable computation, the Transformer is able to retrieve long term dependencies in sequential data. This is enabled thanks to the *attention* mechanism that has been first theorized by Bahdanau et al. [3] and consists in associating weights to each element of the input that represent their importance regarding a specific task. Transformer-based models outperform most of sequential approaches, and in particular LSTM [67] with models like BERT [14] and extensions like RoBERTa [40]. For now,

Transformers are still omnipresent in Large Language Models (LLM) such as GPT [8], LLaMA [66], PaLM, [11] *etc.* It should be noted that these models do not use the entire Transformer architecture. A Transformer has an encoder-decoder structure, which are its two main parts, and BERT is encoder-based whereas modern causal language models are decoder-based.

2.2 Training Procedure

Since we have some deep learning models to use in emotion detection, it is time to describe how the training phase is concretely implemented in a general sense, before tackling our approach specificities in the following parts.

Principle. In order to train and evaluate the model, data first has to be split in 3 subsets called training, validation and test sets. Then all subsets are pre-processed the same way in order to apply the model according to the input format. On the training set, optimization of the model parameters is performed according to the training signal held in the *loss function*. This function is therefore optimized through an optimization strategy described by the *optimizer* (it is often derived from the gradient descent algorithm [59]). Finally, the optimization signal is spread along all the layers of the model going deeper and deeper into it, which is referred to as *backpropagation*.

For instance, if the optimization strategy for the loss function is a gradient descent, the backpropagation algorithm defines the weights update between steps t and $t+1$ as follows:

$$W_{t+1} = W_t + \alpha \frac{\mathrm{d}L}{\mathrm{d}W} \tag{2.1}$$

Where L is the loss function and the parameter α is defined as the *learning rate*. If the model has several layers, the derivative of the loss with respect to the weights becomes a compound partial derivative following the chain rule. Given A and Z two intermediate layers with associated parameters a and z, it can be written:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial a} \frac{\partial a}{\partial z} \frac{\partial z}{\partial W} \tag{2.2}$$

By choosing a loss function and tuning the optimizer's parameters, it is possible to implement and improve several learning strategies, which means that the main challenge is to find the right setup for each application. Once the model is fit on the training set, it is applied to the validation set that contains unseen data to check the relevancy of the fit. In order to gradually improve the parameter tuning, these two steps are repeated several times and are called *epochs*. Once all the epochs are performed, it is time to evaluate the final model on the unseen data contained in the test set. This gives a final evaluation of the model that consists in computing some *evaluation metrics*.

The most common evaluation metrics for classification tasks are *accuracy*, *precision* and *recall*. Accuracy simply consists in computing the percentage of right predictions amongst all predictions. As it is not giving any insight about what cases are correctly or incorrectly predicted, it has to be completed with precision, which accounts for the number of relevant predictions among all predictions for a class, and recall which accounts for the number of relevant elements that are actually retrieved for a class. Then, the $F1$-*score* is a broadly

used metric for classification tasks as it is the harmonic mean of precision and recall. So, it can be seen as a more precise evaluation of the accuracy of the model. In order to formally define them, let's place ourselves in a binary classification scenario (the following can easily be extended to multiclass classification by considering class pairs). Let P and N be respectively the number of real positive and real negative data samples. Let TP, TN, FP and FN be respectively the number of true positives, true negatives (correct predictions of positive and negative classes), false positives and false negatives (incorrect predictions of positive and negative classes). Then, the standard metrics can be defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{P + N} \quad \text{Precision} = \frac{TP}{TP + FP} \tag{2.3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{F}_1 = \frac{2\,TP}{2\,TP + FP + FN} \tag{2.4}$$

In addition to these standard metrics, many other criteria can be computed. In our particular case, we will use Matthews Correlation Coefficient (MCC) [12]. This measures a Pearson correlation [49] between the predicted and the actual class. It appears to be a more challenging metric than the $F1$ score, which gives more precise information on classification quality [4]. Using the previously introduced notations and adding the following:

$$N = TN + TP + FN + FP \ , \quad S = \frac{TP + FN}{N} \text{ and } \quad P = \frac{TP + FP}{N} \tag{2.5}$$

MCC was originally defined in [42] as:

$$\text{MCC} = \frac{TP/N - S \times P}{\sqrt{PS(1-S)(1-P)}} \tag{2.6}$$

Training scenarios. The aforementioned training procedure can serve various training scenarios. The most standard one when dealing with LLMs is to apply *fine-tuning*. Indeed, big models such as BERT are said to be pre-trained, which means that a first training has been performed on very large corpora. From these, the pre-trained model can be retrieved and serve many purposes. Thus, when one wants to use such a model to perform a specific task, it is necessary to adapt it through a lighter training, which is called fine-tuning. Basically, fine-tuning allows to fetch relevant information from the general representation of the pre-trained model. It is usually a task-specific procedure. Then, when considering two tasks that are either similar or connected, meaning solving the second is made easier by learning the first, there exist other *transfer learning* approaches that consists in training scenarios where some knowledge acquired from a source task is used to solve a target task. In addition to avoiding fine-tuning when it is not necessary, transfer learning enables more robust training when we have little data. It is also suitable when one wants to represent some general features, that makes low data training scenarios particularly relevant from a meta-learning perspective.

In particular, the *few-shot learning* scenario is a transfer learning scenario that consists in providing only few examples from a new class (or a new task). This forces the model to make predictions with very little knowledge on the target. For example, it may be

provided with only one example (one-shot), or even none at all (zero-shot). In cases where we want to give the model a small number of examples, we can also follow methods such as N-way k-fold, which involves providing k examples in N different contexts, such as N classes. This is typically used in metric learning, but can also be extended to unseen classes, therefore referring to the zero-shot scenario. Indeed, using some metric learning, the computed distances are relative between the classes, which means that we can theoretically feed the model with previously unseen classes.

2.3 Implementation Details

In order to design and train deep learning models in this project, the selected framework that embeds both pre-built model layers and training tools (loss, optimizers, ...) is PyTorch [48]. In addition to these many available features, this framework especially includes an automatic differentiation engine [47] written in `C++` that allows to retain the parameters gradients and perform backpropagation in a single line of code.

As both meta-learning and conversational data handling are computationally heavy tasks, it is necessary to be provided a large amount of computing power. For that we could use the French HPC network Grid'5000 [5]. It is a testbed for research in computer science, with a focus on parallel and distributed computing, providing researchers access to 15000 cores and 800 compute-nodes in homogeneous clusters in France and Luxembourg. On Nancy site we can have access to GPU clusters from 11GB to 40GB of memory.

Regarding experiments reproducibility, all the code produced so far is available in this Git repository: `https://github.com/B-Gendron/meta_dyda`.

3. Related Work

From now, we need to have a dual focus, since we're looking both at the latest advances in meta-learning methods, and in conversational data processing. This chapter therefore presents the key results of recent literature reviews on both topics.

3.1 A Primer on Meta-Learning

This section provides both an overview and a more specific characterization of meta-learning through its comparison with related fields and a detailed investigation of different application areas. Most of the following section stems from a recent survey on meta-learning [28] and especially uses the same terminology and notations.

The deep learning revolution allowed to evolve from hand-designed features to representation learning, which means that a deep learning model automatically extracts supposedly relevant features. In a similar way, meta-learning allows to evolve from hand-designed learners to learning a learning algorithm. For this reason, meta-learning is often referred to as a "learning to learn" machine learning strategy, which, although legitimate, is not the only manner to describe meta-learning approaches. The following sections aims at broadening meta-learning definition through a comparative study with its related fields. In particular, this work focuses on meta-learning approaches that hold an explicit meta-objective function designed for a single-task application.

In short, meta-learning can be described regarding three aspects expressed as simple questions. Starting with "How?" leads to the definition of the *meta-optimizer* that dictates how the meta-learner should be updated. Then, asking "What?" comes up with the *meta-representation* of the datasets and/or of the tasks. This is the built reference that represents the acquired knowledge from which one can infer. Finally, the question "Why?" finds its answer in the *meta-objective* which guides the meta-learner through the desired task.

Related fields. The following study of the meta-learning ecosystem focuses on training approaches that provide algorithmic frameworks. Nevertheless, one can notice that there are also some modeling approaches related to meta-learning, such as Hierarchical Bayesian Models (HBM). It appears that such models also provide a formalism to describe some algorithm-based models [23], that's why we only focus here on this last category.

At first, a common way to perform representation learning is through the use of Transfer Learning (TL) [72, 71]. This approach involves transferring knowledge from a source task for which data is provided to a target task. It implies parameter sharing and, in most cases, fine-tuning. In vanilla transfer learning there is no meta-learning objective, besides meta-learning can be used to define TL strategies among with other meta-representations. Still without meta-objective, an extension of TL can be seen in Domain Adaptation (DA) or Domain Generation (DG). However, contrary to TL, here the training is performed on a set of tasks represented as a task distribution $p(\mathcal{T})$. It appears that, as well as for TL, meta-learning can be used to perform both DA and DG. Also regarding learning on different tasks, Multi-task learning (MTL) differs from meta-learning in the sense that the objective is to infer on a bunch of defined tasks, without the will to generalize on

unseen tasks. Consequently, this method is not initially using a meta-objective, but there are ways to include meta-objective in MTL to enhance model's stability [38].

Regarding other fields of representation learning, Hyperparameter Optimization (HO) aims at learning an appropriate training setting. As it can be seen as a task for the meta-learner, this family of methods define a meta-objective, such as in gradient-based hyperparameter learning [19].

Finally, there are other techniques that are more marginally related to meta-learning. This is the case of Continual Learning (CL) [13], which is not rigorously part of meta-learning as it has no explicit meta-training objective. However, these two approaches are strongly connected as meta-learning can be used to improve CL baselines [2, 58], and it can also hold requisites for CL in its objective in order to tackle the issue of catastrophic forgetting [46] (*cf. infra*).

A formalism for meta-learning. This part aims at giving more details towards implementation of meta-learning through a relatively light formalism in order to understand the concepts of inner and outer optimization. Let ω be the description of a training strategy. It represents the meta-knowledge used for training and can be an initialization or a hyperparameter selection, for instance. Let $\mathcal{D}$ be the dataset, splitted in some subset such as $\mathcal{D} = (\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{val}}, \mathcal{D}^{\text{test}})$. Then, we need two groups of tasks. The first one is to be used in training, called source tasks, and the second one is to be tested on a test stage, called target tasks. For the moment, let's focus on training stage only, that is to say on M source tasks. Therefore, the considered data is precisely $\mathcal{D}_{\text{source}} = (\mathcal{D}^{\text{train}}_{\text{source}}, \mathcal{D}^{\text{val}}_{\text{source}}, \mathcal{D}^{\text{test}}_{\text{source}})$. Finally, the model training uses the loss function $\mathcal{L}$ and θ contains the model parameters. With such variables, it is possible to formally describe the optimization objectives governing meta-training. It basically consists in two objectives that respectively perform an inner (equation 3.2) and an outer (equation 3.1) optimization.

$$\omega^* = \underset{\omega}{\operatorname{argmin}} \sum_{i=0}^{M} \mathcal{L}\left(\mathcal{D}^{\text{train}(i)}_{\text{source}},\ \theta^*(\omega),\ \omega\right) \qquad \text{outer optimization} \tag{3.1}$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}\left(\mathcal{D}^{\text{train}(i)}_{\text{source}},\ \theta,\ \omega^*\right) \qquad \text{inner optimization} \tag{3.2}$$

Therefore, the first equation represents the meta-training whereas the second one represents the model training. Between these two coupled equations, it can be seen that the inner optimization is designed to provide feedback to the outer, and *vice versa*. As meta-learning is often referred to as a "learning to learn" scenario, this concept can be seen materialized by the outer optimization equation that aims at selecting the best learning configuration. Then, from equation 3.1 that only provides information about meta-training, it is possible to deduce the meta-testing equation that is therefore defined as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}\left(\mathcal{D}^{\text{train}(i)}_{\text{target}},\ \theta,\ \omega^*\right) \qquad \text{inner optimization} \tag{3.3}$$

Given a task i. The meta-testing requires to apply the learned meta-knowledge ω^*, that is why the formula is close to the inner optimization one.

For comparison purposes, equation 3.4 shows how the conventional training setup would be described in this formalism:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{L}\left(\mathcal{D},\ \theta,\ \omega\right) \tag{3.4}$$

Obviously this formalism contains only one equation, and it is only focused on model parameters tuning. Now coming back to meta-learning formalism, it is important to notice that equations 3.1 and 3.2 are not the unique way of explaining meta-learning. Here we consider a procedure based on the "learning to learn" idea, but meta-learning can also consist in learning some meta information that will prove useful for generalization. Both approaches have led to the development of several models, such as Model Agnostic Meta-Learning (MAML) [16] for the first and Siamese Networks [32] for the second. In this work, we will particularly explore the Siamese Network architecture to develop our model.

From this theoretical point of view, let's dive into practical meta-learning, starting by describing the tools that are required to actually implement meta-learning models.

Meta-optimizers. First, it is important to distinguish meta-learning from the usual way of doing machine learning, henceforth referred to as base learning. In such regular setting, the aim is to improve predictions over data instances. In meta-learning, the main focus is towards improving learning algorithms across learning scenarios or episodes. Meta-learners can be described with explicit optimization objective, then they are called optimization-based meta-learning models. In this case, ω represents what we can call the meta-knowledge. This learning can be performed by a neural network for which ω represents the initial weights. It is the case in MAML [16] which leverages gradient-based optimization. However, this exact optimization framework leads to heavy computations due to both inner and outer gradient-based optimization.

In order to get rid of second order derivatives computation, the model-based or black-box approach, often presented as feed-forward model architecture has been proven efficient using networks such as CNN [43] or LSTM. For instance, [56] build a LSTM-based meta-learner to learn an optimization algorithm (based on gradient descent) used by another deep learning model that performs the desired task.

Finally, third way to perform meta-learning in a efficient way is to use metric learning based methods such as Siamese [32] or Matching [69] Networks. These methods actually leverage loss learning which is a sub-domain of metric learning. Both have been proven efficient in few-shot learning setting, even if Siamese Networks seem to perform poorly on one-shot learning according to Vinyals et al. [69]. Another alternative is Relation Networks [64], which is initially designed for few-shot learning but also has a natural adaptation towards zero-shot learning. In our experiments, as we are using Siamese Networks, it is essential to pick a loss from this metric learning framework.

Meta-learning settings and applications. Few-shot learning can be a challenging setting when using deep learning models. Indeed, in this case data volume tends to be a key factor in performance, but training large models with limited datasets, often leads to overfitting or non-convergence of the optimization strategy. This way, meta-learning helps to address this challenge, even though the reached performance do not yet exceed

those of fully supervised approaches [25]. Let's give some examples of meta-learning settings that address the issues of few-shot learning. First, there is Hyperparameter Optimization (HO) which can by applied to an existing meta-learning method in order to improve its performance in FSL. However, HO is not a meta-learning setting in itself. Then, Neural Architectural Search (NAS) can be considered as a type of hyperparameter optimization in which the architecture of a neural network is determined by ω. It is a meta-learning setting but this method is computationally heavy, therefore it is hard to reach optimality still because of non differentiability issues [15, 39]. It is also challenging to find architectures bound to generalize well, *i.e.* that are promising in FSL scenarios. Finally, some Continual Learning (CL) approaches can be used to generalize knowledge across new tasks without damaging too much the knowledge about previous task. This common pitfall in meta-learning, called *catastrophic forgetting*, can therefore by address by such methods.

Issues According to Hospedales et al. [28], even if there are several ways to define the meta-knowledge ω, a definition with too many parameters would not be a scalable solution. Moreover, there are some issues with gradient descent meta-optimization that can appear in two ways: the first one would be if the meta-objective is not differentiable, in which case alternative approaches may be suitable like reinforcement learning of evolutionary algorithms. The second one is because of a gradient-based inner optimization, outer optimization computations are costly as they require a second order derivative.

Challenges and further works. One of the main challenge is to broaden the definition of the task distribution $p(\mathcal{T})$. In can be for example a multi-modal distribution or more generally involve different learning strategies to be trained on. Furthermore, the objective is not only to be able to learn relevant features but also to generalize properly across a variety of tasks. Here, the domain shift may be an issue to handle. In ERC, one may also want to generalize across labels, therefore being able to compute predictions on previously unseen labels. By using a meta-learning approach in this work, we expect that, in the end, our final model will be capable of such generalization.

There are also some computational challenges that explain why the FSL setting is often preferred for it is less costly than a fully supervised approach. Moreover, computing second order gradients can be harmful in such cases, that's why there exists some tricks to avoid full differentiation. In this perspective, feed-forward models have been proven efficient.

3.2 Deep Learning Methods on Conversation Data

In this section, we focus more on conversational data handling concerns. Precisely, the aim of this work is to perform emotion prediction on conversational data, which is a particularly specific type of text data. The following will explore common practices and challenges regarding conversational data in the specific scope of emotion recognition. In that case, such a task is often referred to as ERC, which stands for Emotion Recognition in Conversation. Its principle is very similar to the one of sentiment analysis which is one of the most famous NLP tasks, even if sentiment and emotion are rigorously two different concepts. Thus, this section stems from a recent survey on textual conversations [50] and gives insights about the latest advancements and challenges in the use of deep learning

approaches for ERC. These insights will be particularly useful for our experiments, not only to build an accurate context representation but also for qualitative interpretation of the results.

Conversation and emotion representation. This first concern is about the conversation itself. A conversation has a context, which means that it is necessary to use appropriate utterance representations that account for information in preceding and following utterances. Moreover, in real-life conversations, some useful contextual elements can be underlying or implied but not expressed as words in the text. Some parts of the explicit context can also be expressed through common sense expression or informal language.

Regarding emotions, in the following work we are going to focus on dyadic conversations where each speaker turn (i.e. each utterance) is labelled with one emotion. This way, it is not necessary to store the identity of the speaker for each emotion expressed and the retrieved information is more precise than a global feeling for the whole conversation. Indeed, no matter how one stores the emotions, it is important to account for emotion shifts through statements since the emotions conveyed during a dialogue can change. In addition, it is possible to find causality between emotions which can explain why such emotion is expressed : since A got angry at B for his behavior, B may feel offended when he would not have been without A's words.

Conversation Understanding. In order to produce accurate and relevant predictions on ERC, it is necessary to ensure a right and complete understanding of the conversation along with its emotions. The first difficulty in that sense is ambiguity that lies behind emotion annotation. This can induce bias in training, then consequently in evaluation [20]. Still regarding annotation, there are multiple ways to represent emotions that lead to different training and inference settings. The most broadly used setting is the categorical approach which boils down to a simple labelling procedure. Furthermore, in this setting it is possible to account for mixed emotions with explicit annotations thanks to multilabelling. On the other hand, emotion can be described with a dimensional approach [17], as a weighted combination of different aspects. This is also an appropriate design for mixed feelings but seems difficult to implement as a labelling system, in addition to be very costly.

Moreover, another crucial aspect in conversation understanding is the emotion distribution in the dataset. Indeed, if the dataset is not well balanced, which is often the case due to the natural abundance of neutral utterances, the model mostly learns to predict the absence of emotion instead of a spectrum of different feelings. This is an issue in our case as the dataset we use is very unbalanced. We describe in next chapter how we tried to mitigate this issue.

Annotation subjectivity. Allocating an emotion to a sentence is obviously a subjective task, since humans are evaluating content produced by humans. In order to achieve the most reliable annotation of the dataset, there are two approaches. The first assumes that a ground truth label exists. In this case, several experts annotate the sentence and the emotion is chosen either by majority voting or by weighting the experts' evaluations by scores. The second assumes that there is no ground truth label. In this case, light annotation is used, assigning to each sentence a distribution of labels weighted by their

probability in such a context. Annotation subjectivity is one of the aspects to be explored in qualitative analysis of results.

Overview of Available Datasets. Regarding conversational tasks, data availability is often a challenging point. There are actually very few accessible datasets and not all of them consist in real-life conversations. Here are the main datasets that can be encountered in conversation-related deep learning works. For each dataset, if possible, we indicate state-of-the-art performances in ERC, whether it uses meta-learning or not.

- IEMOCAP [9] is the main baseline for multi-modal emotion recognition tasks. In particular, it contains more than 7,000 utterances from dyadic dialogues. These are real-life conversations that aims at being realistic and spontaneous. Li et al.[36] propose a new model structure adapted to multimodal emotion detection that achieves 69.49% accuracy on textual data only, without using meta-learning.

- MELD [52], EmoryNLP [73] and Friends [29] are three datasets containing scripted dialogues extracted from *Friends* TV show. Even if these conversations come from actual discussions between humans, they are not spontaneous and may not be very realistic. On MELD dataset, Song et al. [63] leverage Prototypical Networks [62] along with contrastive learning [34] to perform ERC and achieve 67.25% in weighed F1 score. The same model on EmoryNLP achieves 40.94% in the same metric.

- DailyDialog [35] consists in more than 12,000 generated dialogues intended to be representative of daily concerns. These dialogues are labelled with the 6 Ekman's emotions. This is the dataset we are going to use in the experiments. Liang et al. [37] propose a model based on Graph Neural Networks (GNN) and CRF that achieves 64.01% in micro F1.

- EmoContext [10] and EmpatheticDialogues [55] contains dyadic dialogues between a human and a conversational agent, these are thus semi-generated conversations. EmpatheticDialogues provides twice as much conversations that DailyDialog. On EmoContext, Ragheb et al. [54] achieve 76% in micro-F1 by using an attention-based model, therefore no meta-learning.

Now that the available tools and the points of attention for doing ERC are given, a question remains about how to develop a deep learning model able to detect emotions in conversations. There is first a need to explore the constraints on the utterances representations to be given to the model. Afterwards, we will see what kind of models can be used according to the specificity of this task.

Using embeddings. *Contextual embeddings* are a type of word embedding model that incorporate context from surrounding words in order to generate representations of words that capture their meaning within a given context. This implies that the word representation will differ depending on the context, contrary to static embeddings. Such embeddings allow for improved performance on ERC where the context conveys many clues to properly understand the expressed feelings. When building our context-aware model, we use attention-based contextual embeddings on the whole dialogue in order to highlight any relevant contextual information.

Designing models. When designing deep learning models for ERC, the main concern is to have an accurate and complete contextual representation. The need to account for sequential dependencies would point to RNN. Even though, since it may be necessary to retrieve information far away from the target utterance, a model architecture like LSTM networks would be more adapted as it accounts for long term dependencies. As the contextual window of an utterance is actually two-sided, symmetrically, a Bidirectional LSTM (BiLSTM) seems to be relevant as used in [30].

As bidirectionality still suffers from lack of understanding long term dependencies, attention-based approaches, such as Transformers, has been proven efficient on ERC. More precisely, it uses Transformer encoder on the conversation, therefore leveraging self attention. Finally, instead of encoding the content of the conversation, it is possible to encode dependencies and relationships between speakers and utterances using graph-based approaches. However, as it consists in exploring the neighborhood, weaker relationships tend to be ignored, that's why such approaches are sometimes combined with recurrent methods in order to still account for long term dependencies [41].

4. Context-Aware Meta-Learning

Now that we've reviewed all the essential prior knowledge about meta-learning algorithms for ERC, let's consider the technical aspects of our implementation, along with the results. The first part of this chapter is devoted to describing the available data. This is followed by a detailed description of the experimental setup, and then both quantitative and qualitative analysis of the obtained results.

4.1 Data

Description. Amongst previously presented datasets, the first selected dataset for the experiments is DailyDialog [35]. The main advantage in using DailyDialog at first sight is that it is relatively small, therefore it is quite easy to handle the entries and run tests on it. However, it as two main shortcomings. Firstly, one has to notice that the dataset is highly unbalanced because many of the utterances are labelled with `no emotion` label. This has to be considered during model training in order to learn also on actual emotions. Secondly, all the conversations of DailyDialog have been artificially generated, which has to differ from conversations of the real world. Therefore, even if the experiments further described in next sections have been held on DailyDialog, one of the main perspectives of this work is to be able to test on other datasets (such as EmpatheticDialogues [55]), or at least to account for a satisfying transferability of acquired knowledge, especially on human-to-human conversations.

Overview. More precisely, Figure 4.1 provides an overview of the dataset. It is already splitted in `train`, `validation` and `test` sets (see Figure 4.1a). Regarding each entry, it contains the dialog content, the dialog acts and the dialog emotions (see Figure 4.1b). A dialog act characterizes what a speaker is trying to express in its speaking slot. This will not by studied in the following as we are going to focus on emotion labels only.

```
DatasetDict({
    train: Dataset({
        features: ['dialog', 'act', 'emotion'],
        num_rows: 11118
    })
    validation: Dataset({
        features: ['dialog', 'act', 'emotion'],
        num_rows: 1000
    })
    test: Dataset({
        features: ['dialog', 'act', 'emotion'],
        num_rows: 1000
    })
})
```

(a)

```
{'dialog': ['Say , Jim , how about going for a few beers after
dinner ? ',
  ' You know that is tempting but is really not good for our
fitness . ',
  ' What do you mean ? It will help us to relax . ',
  " Do you really think so ? I don't . It will just make us fat
and act silly . Remember last time ? ",
  " I guess you are right.But what shall we do ? I don't feel
like sitting at home . ",
  ' I suggest a walk over to the gym where we can play singsong
and meet some of our friends . ',
  " That's a good idea . I hear Mary and Sally often go there to
play pingpong.Perhaps we can make a foursome with them . ",
  ' Sounds great to me ! If they are willing , we could ask them
to go dancing with us.That is excellent exercise and fun , too .
',
  " Good.Let ' s go now . ",
  ' All right . '],
 'act': [3, 4, 2, 2, 2, 3, 4, 1, 3, 4],
 'emotion': [0, 0, 0, 0, 0, 0, 4, 4, 4, 4]}
```

(b)

Figure 4.1: Some details about DailyDialog organisation (left) and entries format (right)

4.2 Experimental Setup

Model choice. From now, regarding the overall model we will focus on Siamese Network architecture which has been identified to be a promising model in such context. Compared to other metric learning approaches, Siamese Networks has been chosen also for pedagogic purposes, as its structure is relatively simple, and also to compare with Prototypical Networks which has already been evaluated on related tasks [24].

Evaluation criterion. In order to evaluate Siamese Networks, a common way is to use the triplet loss which purpose is to grant greater similarity to elements of the same class. This can be done by comparing the distance between entries of the same class and entries of different classes (considering ground truth labels). Formally, one defines a triplet of entries that contains an `anchor`, a `positive` and a `negative` entry. `anchor` and `positive` share the same class whereas `negative` is from another one. For concrete implementation, this setup is broaden to multiple entries in a row. Therefore, let's consider a triplet of batches (A, P, N) where all entries in A and P belong to the same class, and all entries in N belong to another class. The triplet loss can be defined as follow:

$$\mathcal{L}(A, P, N) = \max\{d(a_i, p_i) - d(a_i, n_i) + \text{margin}, 0\}$$

The margin parameter is here to ensure that a relevant optimal solution is found. Otherwise, the optimal setting would be to simply set all the distances to zero, which is not a suitable solution. One usually sets this parameter to 1.

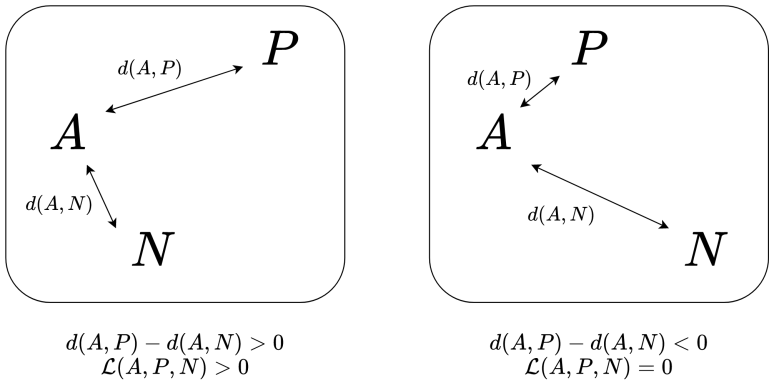


Figure 4.2: Triplet loss value depending on the configuration between anchor, positive and negative samples.

Specificities in training. Due to the use of the triplet loss, the model needs to have 3 parallel forward paths to encode a triplet where similar entries should be aligned compared to dissimilar ones. The main advantage of this setup is that the triplets are built by randomly selecting the two represented classes. This way, even if it conduces to little oversampling, the loss is applied to a more balanced dataset than the original one which reduces bias. However, this training approach sometimes has trouble converging. Therefore, in order to reach optimality more easily, in particular when considering conversational context, it has been considered to use some warm-up training steps. It consists in 3 epochs performed with a very small learning rate which allows to slightly move

in loss landscape in order to more easily reach a global optimum during main training. In addition, accounting for the fact that having an accurate learning rate is essential in meta-learning, it as to be relevantly chosen, either through an empirical study, or by using standard training or optimizer strategies such as decay rate or learning rate scheduler.

4.3 Experiments

Based on this setup, we carried out two main experiments: the first involved producing emotion predictions without taking the context into account, and the second considers a way of representing this context that can be included in the model. Let's describe these two approaches separately.

4.3.1 Siamese Networks on Isolated Utterances Representations

In order to gradually increase the complexity of the implementation to reach the desired architecture, the first step is to consider *isolated utterance* representations, *i.e.* that does not take into account the conversational context. These may also be referred to as *static utterances*, in reference to static embeddings, in the sense that the representation of the utterance remains the same whatever the dialogue it belongs to. This first step is useful to alleviate the preprocessing step at first. Such static representations are obtained through tokenization using FastText encoding, which means that each word is represented by its index in FastText vocabulary. From this prepared data, the dataset has been formatted to form triplets in order to fit with triplet loss computation. Regarding the core model of the Siamese architecture, we experimented with different models to evaluate the influence of core model choice. The overall training pipeline, including the specific data handling is illustrated on Figure 4.3.

4.3.2 Siamese Networks on Conversation-Aware Utterances Representations

Once this preliminary work is done, the idea is to adapt this first model to context-aware representations. For this purpose, we selected an attention-based encoding at the dialog level. To enable such attention we chose Transformer-based architectures, especially BERT models. Before applying the model, it is necessary to use the associated tokenizer that contains special tokens such as `[SEP]` which will prove very useful as it indicates the separation of utterances. To have a better idea of how BERT tokenizer acts on text, an end-to-end example is provided in Appendix in Figure 5.4. A BERT tokenizer typically converts the text of one dialog into up to 512 tokens, including `[SEP]` tokens. This size limitation is an important concern at this stage because it implies that a part of the dialog will be lost. When performing tokenization, it is possible to set either left or right truncation. Thus, some probing of DailyDialog emotion labels has been done in order to have an idea of the emotion distribution over all dialogues. As shown in Figure 4.4, it appears that the beginning of the dialog seems to convey more emotions that the end. A more precise study as been held over each emotion (see Figure 5.1 in Appendix) and leads to the same conclusions. For this reason, the tokenization will be performed using right truncation.

So far we have described the whole tokenization process, which corresponds to the first

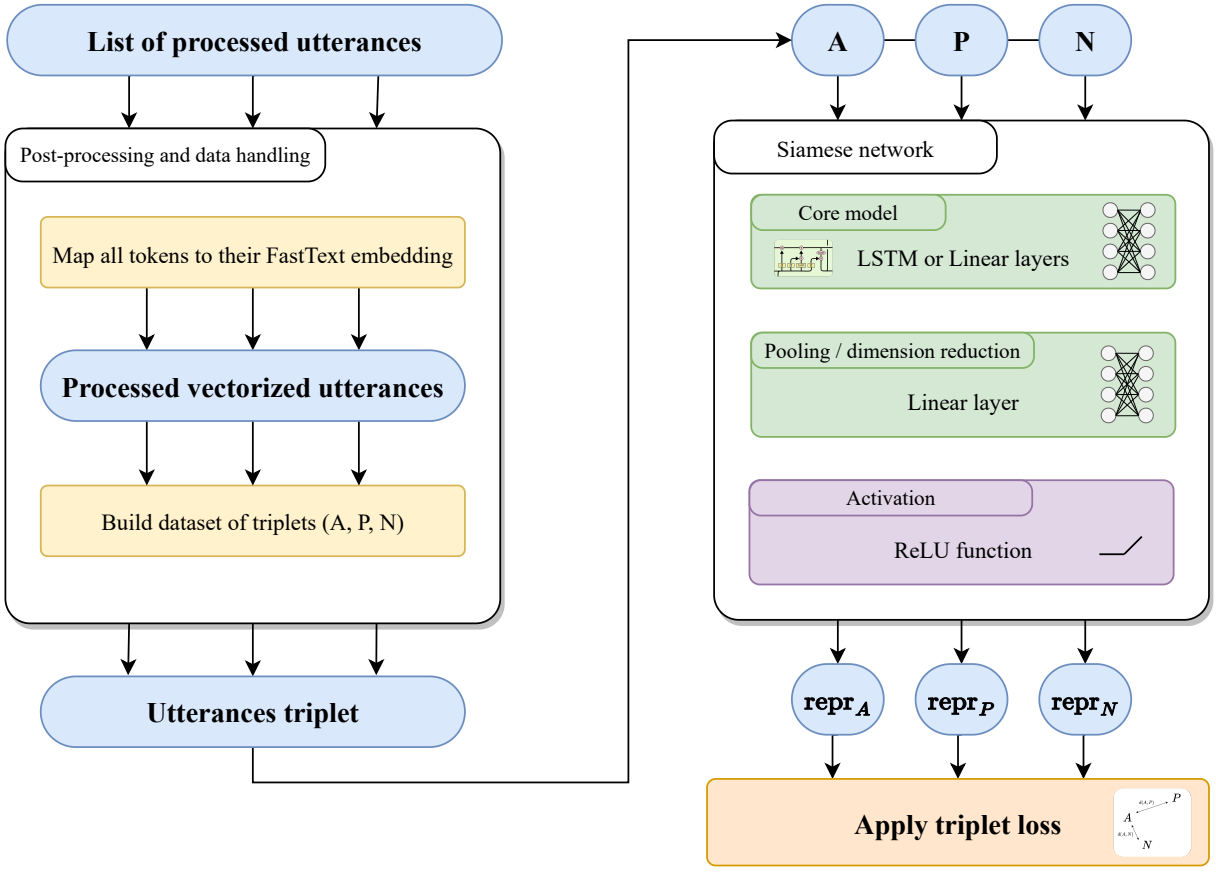


Figure 4.3: Training pipeline for emotion recognition on static utterances representations using Siamese networks.

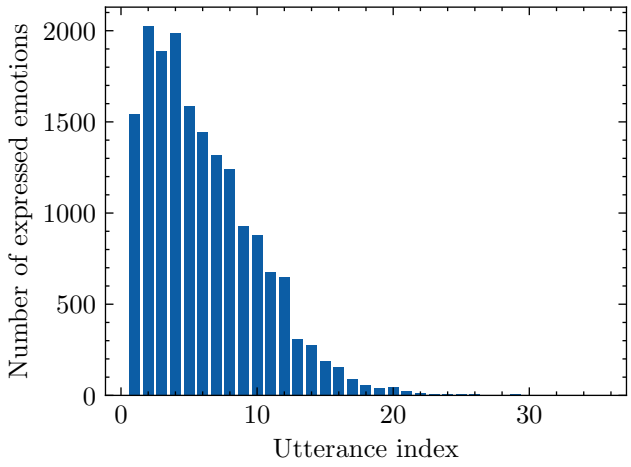


Figure 4.4: Distribution of expressed emotions *w.r.t.* the utterance index in dialog

three steps in Figure 4.5. The next step is to encode the dialog using BERT model, in order to get a contextual representation of the conversation. Thus, each token is encoded into a 768-dimensional vector. The dashed brackets are not actual splits, they are just there to identify what will be used to form the contextual representations of utterances. Similarly, the `[SEP]` token also has its own vector encoding, which is not shown here for clarity. Since the boundaries of each utterance are saved, it is easy to split this BERT output accordingly. The last step is finally to flatten and pad each utterance representation using

a linear layer which outputs a 512-dimensional vector (512 being the padding length).

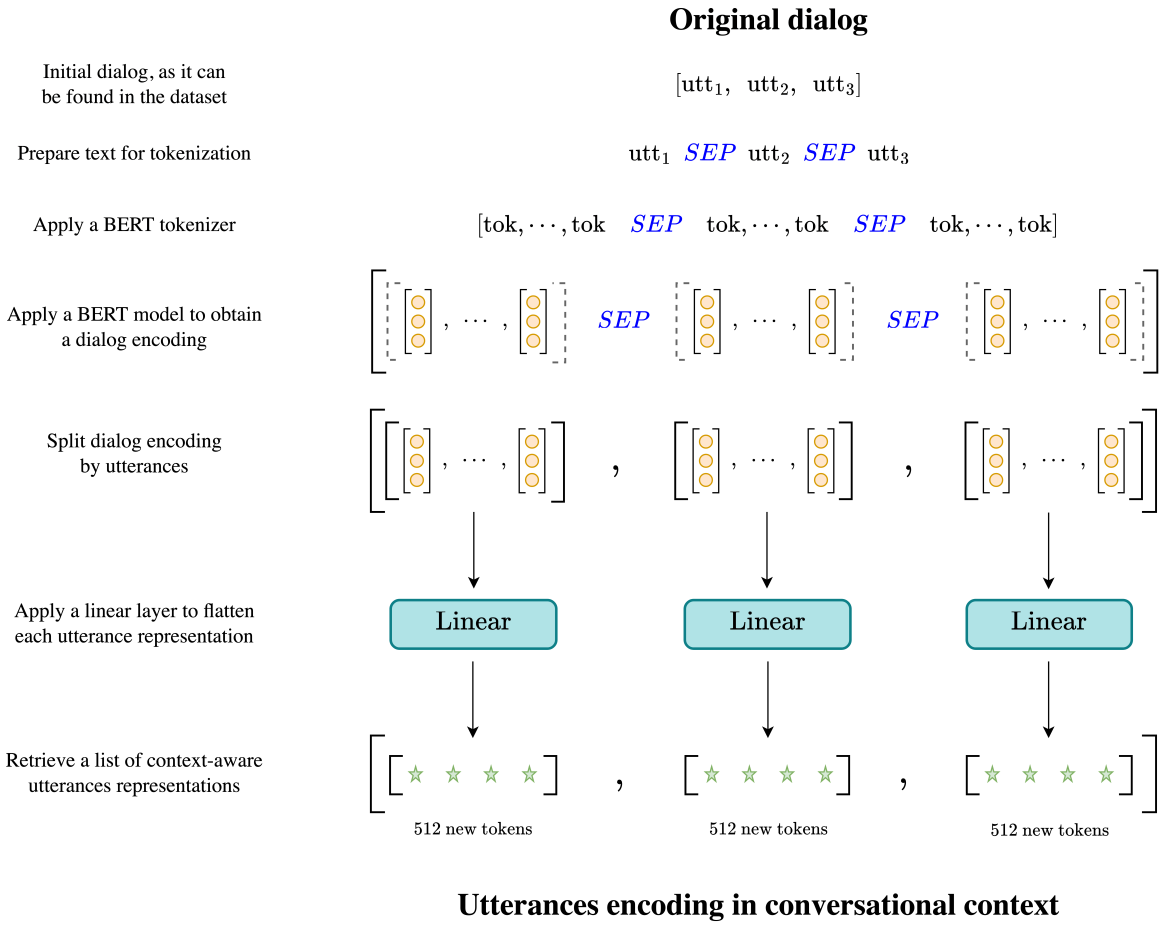


Figure 4.5: Description of the whole preprocessing and encoding process from dialog text to conversation-aware utterances representations.

Thanks to the aforementioned process, one ends up with context-aware representations of the utterances, that we can call conversation-aware in this particular context. From such representations it is possible to apply triplet loss on top of it to learn the utterances emotions. The overall model is still a Siamese Network in which the core model would be the encoder used to generate context-aware representations. The detailed process used to train such model is described on Figure 4.6. Indeed, the first step is to train the encoder named `EmCoBERT` for our experiments. From this BERT-type encoding, we first train an emotion classifier on CE loss and then use the encoded representations to train on triplet loss. The underlying hypothesis is that this training procedure will lead to an encoding which is favorable to contextual emotion detection.

4.4 Results

Now that we have a complete description of both experiments, let's have a look at the results on the test set. First, we'll look at the results from a quantitative point of view, by evaluating classification metrics. Next, we'll take a closer look at the predictions in order to provide a qualitative analysis.

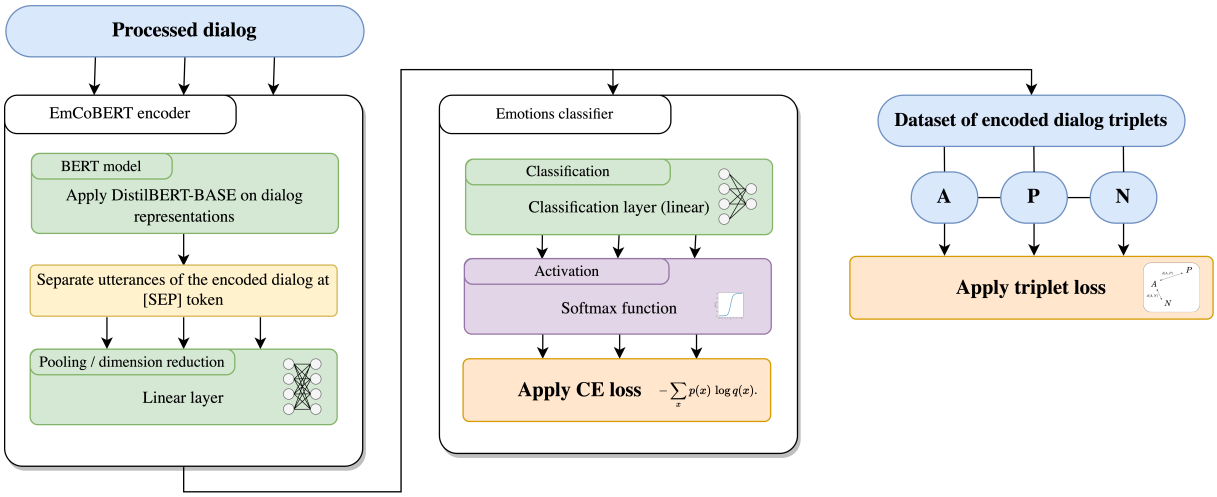


Figure 4.6: Training pipeline for emotion recognition on contextual utterances representations, using Siamese Networks.

4.4.1 Quantitative Analysis

Regarding the quantitative evaluation of both models (isolated utterances and contextual), three main experiments have been conduced. One on the isolated utterances model only in order to find the most adapted core model, one on the contextual model only to find the more appropriate training setup, and finally one joint experiment on both model to compare their performances once they are accurately tuned.

Isolated only. Regarding isolated utterance representations, two main setups have been considered: LSTM and Linear layers (Multi Layer Perceptron, MLP) as core model. The values of several classification metrics on test set are presented in Table 4.1. It appears that LSTM-based models achieve overall best performances than MLP-based models, which is certainly due to their ability to capture some contextual information within the utterance. However, it is best not to use too many layers of LSTM, in which case one observes a decrease in performances. In fact, by adding layers that consider the entire utterance (thanks to long-term memory), we tend to end up with an over-encoded representation, and therefore lose specific elements that support certain emotions.

Contextual only. Regarding conversation-aware utterance representations, the first experiment has been held on a subset of the training set containing either 1000 or 5000 samples (which means up to half the training set), in order to run several experiments in a reasonable amount of time. This reduced training set is enough to compare different setups in terms of hyperparameters and select, in particular, an appropriate learning rate. Thus, it is worth noting that in the following, the learning rate has been chosen after an empirical study. Methods such as decay rate and/or learning rate scheduler are yet to be tested. Then, the batch size has been fixed to 8, which seems small but is actually the maximum possible regarding memory constraints. Table 4.2 gathers some of the results for this set of experiments.

As it is usual in meta-learning, it seems that the results are sensitive to the learning rate. Moreover, even if the results are not that good at this stage, it is interesting to notice that the last run, using five times more training data than previous runs, achieves

Layers	Accuracy↑	Loss↓	Precision↑	Recall↑	wF1 score↑
LSTM-based models					
3	68.0	**0.706**	0.682	0.680	0.680
4	67.9	0.734	0.678	0.679	0.677
5	**70.0**	0.711	**0.702**	**0.700**	**0.700**
6	65.4	0.756	0.658	0.654	0.655
7	66.2	0.793	0.664	0.662	0.661
MLP-based models					
3	**68.4**	**0.747**	**0.685**	**0.684**	**0.684**
4	62.9	0.834	0.629	0.629	0.628
5	64.7	0.801	0.650	0.647	0.647

Table 4.1: Main results using Siamese Networks on static utterances representations. Best values are in **bold** and arrows indicates if greater or lower is better.

Tr. samples	L. rate	Epochs	MCC↑	Loss↓	Precision↑	Recall↑	wF1 score↑
1000	5e−4	10	0.422	5.56	0.506	0.505	0.505
1000	1e−3	5	**0.456**	**4.26**	**0.535**	**0.534**	**0.534**
1000	5e−4	5	0.412	5.54	0.497	0.496	0.496
1000	1e−4	5	0.435	5.82	0.516	0.516	0.516
5000	5e−4	3	0.454	5.31	0.532	0.532	0.532

Table 4.2: Results for Siamese networks on conversation-aware representations in different configurations.

results almost equivalent to the best run, even though it is not performed in the optimal configuration. This strongly encourages to run experiments on the whole dataset. Finally, the setting of five epochs have been proven relevant in most cases and is retained for the following.

Isolated *versus* contextual. The last experiment described in this part aims at comparing the static and the contextual approaches, both in their optimal configuration. Then, as in meta-learning we study the ability of a model to transfer knowledge, it is crucial to ensure stability through a small variance. This is therefore a relevant criteria at the time of evaluation, in the sense that a less efficient model with a lower standard deviation will be preferred to a model with a large F1 score that produces highly variable results. To actually compute the variance in the following results, predictions on the test set have been performed 10 times using the same trained model, then the metrics have been averaged and the variance is given from these runs. In the results presented in Table 4.3, each experiment has been made twice, for two different learning rates. The first thing we can observe is that the isolated utterance model outperforms the contextual model by about 17% in weighted $F1$ score in both cases. This suggests that a contextual model with a Transformer-based encoder is far more challenging to tune than a LSTM-based model on isolated utterances. Regarding the variance, it seems fairly equivalent from one model to the other. In general, both models are less stable with a learning rate

of 1e−4 rather than 5e−4.

QUANTI-STAT		
Avg. on 10 runs	`lr` = 0.0005	`lr` = 0.0001
Loss↓	$0.7821 \pm 1e{-2}$	$\mathbf{0.7652 \pm 1e{-2}}$
MCC↑	$\mathbf{0.6373 \pm 4e{-3}}$	$0.6243 \pm 6e{-3}$
Precision↑	$\mathbf{0.6874 \pm 4e{-3}}$	$0.6760 \pm 5e{-3}$
Recall↑	$\mathbf{0.6883 \pm 4e{-3}}$	$0.6777 \pm 5e{-3}$
wF1 score↑	$\mathbf{0.6843 \pm 4e{-3}}$	$0.6759 \pm 5e{-3}$
QUANTI-CONV		
Avg. on 10 runs	`lr` = 0.0005	`lr` = 0.0001
Loss↓	$\mathbf{5.2352 \pm 0.005}$	5.8808 ± 0.1610
MCC↑	$\mathbf{0.4286 \pm 6e{-3}}$	$0.4238 \pm 1e{-2}$
Precision↑	$\mathbf{0.5103 \pm 5e{-3}}$	$0.5064 \pm 1e{-2}$
Recall↑	$\mathbf{0.5102 \pm 5e{-3}}$	$0.5061 \pm 1e{-2}$
wF1 score↑	$\mathbf{0.5111 \pm 5e{-3}}$	$0.5060 \pm 1e{-2}$

Table 4.3: Table of results on test set

4.4.2 Qualitative analysis

Prediction distributions. Regarding predictions on isolated utterance representations, all graphs are represented in Figure 4.7 to give an overview of the prediction distribution, label by label. The equivalent graphs for predictions on contextual utterance representations are given in Figure 4.8. Each pie chart gives the distribution of predictions when the true label is the one written below the chart. For easier reading, each pie chart is available individually in the Appendix (Figures 5.6 and 5.8).

What we can first observe for isolated representations is that, except for `fear`, the expected emotion is always the most often predicted, in the sense on an absolute majority. According to the overall label distribution given in Figure 5.2, it seems that the best predicted labels are also the most frequently encountered. On the contrary, the rarest labels which are `fear` and `disgust` are those with the lowest proportions of correct predictions. These two trends are also observed in the case of contextual representations, except that, in this case, we have a relative rather than an absolute majority in four cases out of seven.

Then, regarding wrong predictions, in both cases we can observe that they are equally distributed between the remaining classes. This is due to the use of triplet loss and in particular the formation of triplets, which involves randomly selecting the classes present in the triplet according to a uniform distribution. Since this selection is not weighted with respect to class frequencies, in the end the labels are equally represented.

In-depth prediction analysis For this qualitative study, we focus only on predictions made on contextual representations. Starting from the contextual representation of the dialogues, we selected only the dialogues where it is possible to build a triplet of utterances, in terms of triplet loss. Then, by indexing the whole test set, it is possible to retrieve the original dialog text associated to each prediction. From this setup, we were able to

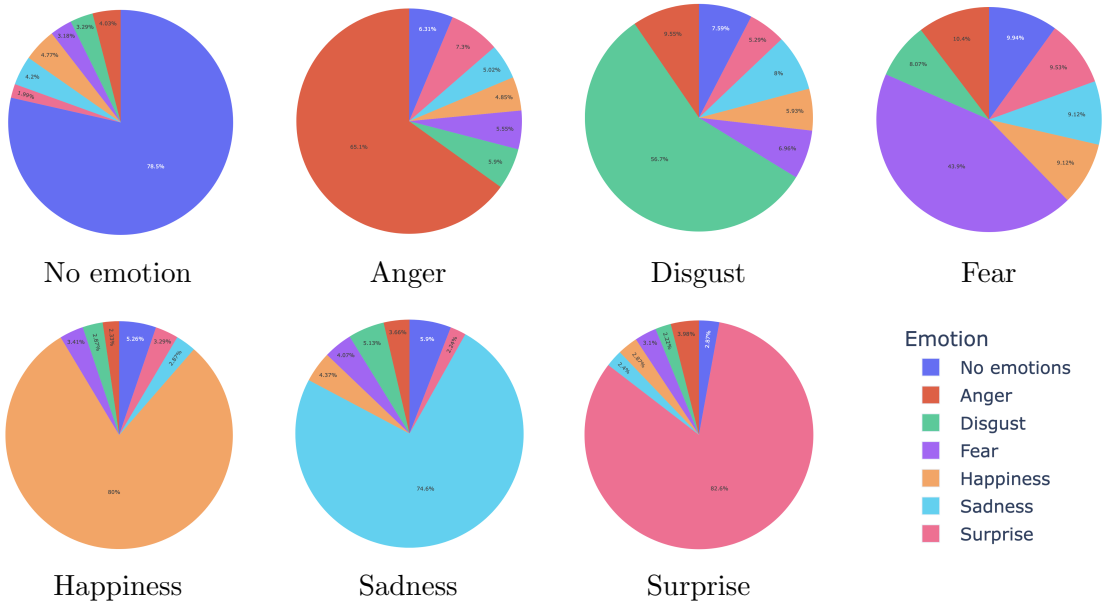


Figure 4.7: Distribution of predictions for each actual emotion in the case of isolated utterances representations.

Figure 4.8: Distribution of predictions for each actual emotion in the case of contextual utterances representations.

extract and look into predictions on 495 dialogues, which represents approximately half of the test set. In the following, each cited sentence is an utterance or part of an utterance that actually belongs to the test set. In addition, any label other than `no emotion` will be referred to as an *emotional label*.

A first finding that is quite common in such experiments is that misclassifications are not random. Thus, whereas in a previous work on Prototypical Networks [24] it was found that `surprise` was often wrongly predicted compared to other emotional labels, here we find that it is `happiness` that fulfills this role. Indeed, it is very often predicted instead of `no emotion` in such situation that we can imagine the prediction most of the time relies

on anchor words (in the sense of stumbling blocks, nothing to do with the anchor from the triplet loss). They include words like "fine", "really", "all right", "great", "yeah", "cute" or "terrible". These words often carry emotion, but not in every context. For instance, "great" can be used to express joy, but also to mean an agreement, which can be purely informative. Additionally, "nice" can be used when the protagonist is happy, but also in some greetings like "nice to meet you". Therefore, the presence of these pivot words will often lead to a wrong prediction in favor of an emotional label (mainly `happiness` and, less often, `surprise`) rather than the `no emotion` label.

Another factor that leads to misclassification seems to be the presence of some punctuation marks that may be associated with emotions such as "?", "!" or "...". Here we are still pointing out cases where `no emotion` is to be predicted. Indeed, in many neutral utterances, the presence of such prediction marks leads to an emotional prediction, most of the times being `happiness`. Nevertheless, it is also possible to observe a correct prediction of neutrality even when such punctuation marks are present, and this happens especially when the context of utterance is rather neutral. This is an important finding because it brings situations when the conversational context seems to influence the nature of the prediction. Nevertheless, if we compute predictions for the same utterances with isolated utterances model, it appears that these are correctly predicted as well. Therefore, we have to be careful when interpreting such contents. Such situation is illustrated by the example in Figure 4.9a. The utterance in itself seems quite emotional with the expression "Oh no!" and the presence of punctuation marks. But if we look more closely at the context, we end up with a common situation of a person looking for their way, hence the "no emotion" label.

Back to the `happiness` class, we realize that it is predicted for many basic, factual questions such as "When do the playoffs start?", "How much is it?" or "That's all I have to do?". Therefore, we end up with a broadly predicted label, sometimes in irrelevant contexts. To be more precise, `happiness` is almost always correctly predicted when it comes to be the right class, consequently this particular label has a very good recall, but a very bad precision.

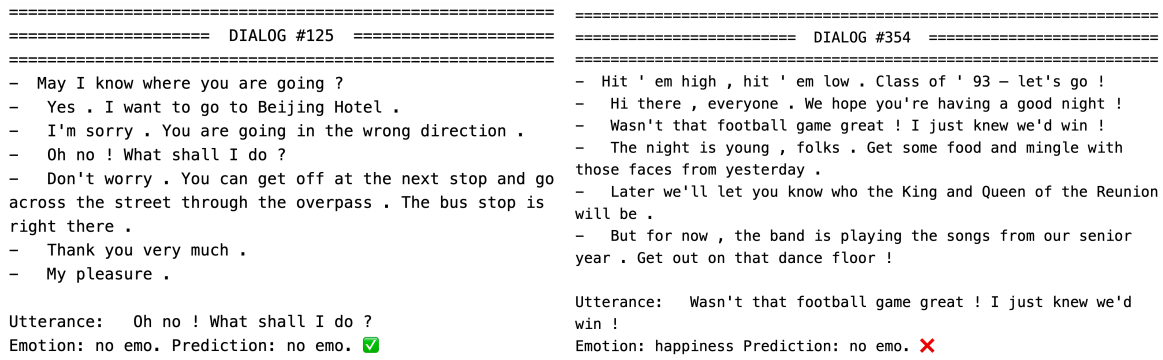
Overall, apart from the aforementioned cases, `no emotion` class is often accurately predicted when the sentence is either declarative or informative. It works particularly well when the utterance does not contain any adjective, such as "Here you are." or "And the bubble wrap?". This is overall a quite expected finding as `no emotion` class is broadly represented in the dataset.

Apart from these two most common labels, let's focus on the other emotional labels. First, we can discuss the case of `fear` and `disgust`. They appear to be quite similar labels in the sense that they are usually correctly detected in obvious situations such as "I'm really in a flap about the interview." for `fear` or "It looks like some kind of primitive form of torture." for `disgust`. However, these labels also appear wrongly in some neutral contexts such as "The service did not help the situation." (emotion: `no emotion`, prediction: `disgust`), which seems very surprising. In fact, according to the emotion labels distribution across the datasets (*cf.* Appendix 5.2), these emotions are poorly represented, which is probably why these classes are misunderstood.

Then, regarding `sadness` and `anger`, we notice that they are often predicted in presence of words expressing strong feelings such as impossibility or urgency for `anger`: "I can't send out mails. We'd better call the IT department and ask them to check it immediately.".

For `sadness`, we often find negative or restrictive words such as: "No, thanks. I've had more than enough. In fact, I must be running along.". Same as for `happiness`, such words seems to play the role of anchor words which is not always leading to correct predictions, as it is the case in the aforementioned examples.

Overall findings. As seen in the previous label-by-label study, it seems that `no emotion` label can be correctly predicted on an utterance with emotional markers when its context is overall neutral. This suggests that context does help sometimes, but it can also be found situations of wrong predictions when context should have helped. An example is given in Figure 4.9b. Here, the utterance in itself expresses happiness but it has not been detected. Once again if we look closely at the context, we see that not only this utterance but rather the whole dialog reflects a happy atmosphere. This should have helped the conversational model to accurately predict this emotion. In fact, it seems that these wrongly predicted utterances for which context should help are particularly difficult to predict in general. Indeed, when inferring with isolated utterance model on such examples, most of the predictions are wrong too.

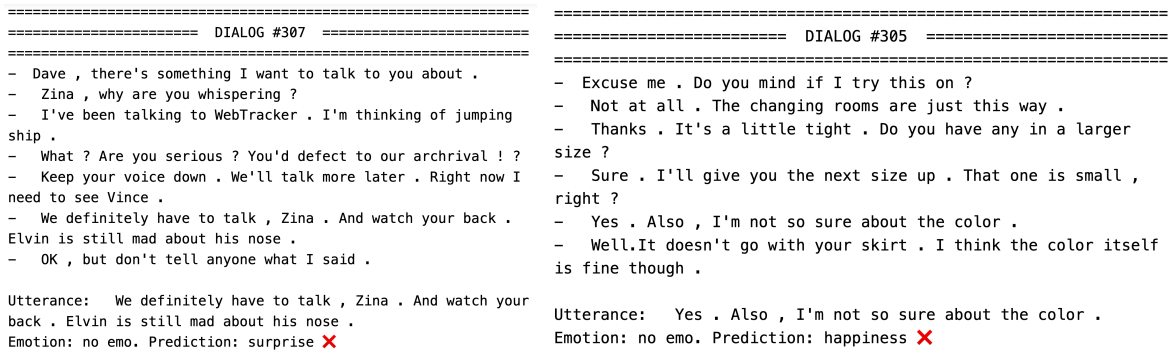


(a) (b)

Figure 4.9: Two examples of situations where the context seems to influence the prediction. On the left, the context seems to have helped providing the right prediction, whereas on the right it seems to have not been taken into account.

Moreover, we found several situations when the conversational model wrongly predicted an emotional label instead of `no emotion`. A closer look at the utterances corresponding to these examples reveals that they often convey an emotion, even if it cannot be described by one of the 6 emotional labels, maybe due to their low intensity. This suggests that the model has learned a certain sensitivity to the concept of emotion, without always being able to accurately describe it. Indeed, the assigned emotion is not always consistent with what is expressed in the dialogue. This is typically what can be observed in the examples shown in Figure 4.10. In these examples, not only the utterance in itself but also the context would suggest that a feeling is conveyed, even if it does not correspond to any of the six emotional labels. Indeed, in most of the cases, it seems that such emotional predictions are triggered from the conversational context. To add a quantitative insight, let's consider all samples labelled with `no emotion`. If we compute the ratio of emotional predictions over `no emotion` predictions, we end up with 0.27 on the isolated utterances model, and 1.04 on the contextual utterances model. Therefore, the contextual model is very likely to predict emotion when the label is `no emotion`, contrary to isolated utterances model. Overall on all labels, this trend is present but less pronounced for all labels, with 17% of

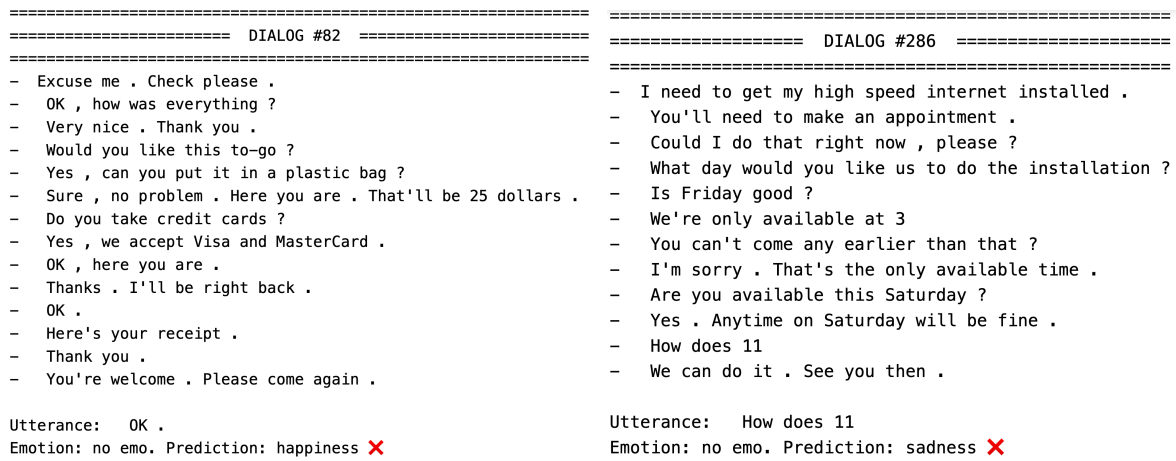
"no emotion" predictions for isolated utterances, versus 15% for the contextual model.



(a) (b)

Figure 4.10: Two examples of situations where the model predicts an emotional label instead of `no emotion`. As it can be seen, the predicted emotion does not always correspond to the feeling conveyed in the dialog.

Critical aspects. First, it appears that there are still some unexplainable situations, that is to say where neither the utterance nor the context help to explain the output. This point is not surprising as explainability is often at stake when dealing with deep learning models, also it seems to represent a minority of examples among the samples. Two of them are given in Figure 4.11. For both of them, the context seems overall neutral, and the predicted emotion does not correspond neither to the utterance itself, nor to any other utterance in the dialogue.

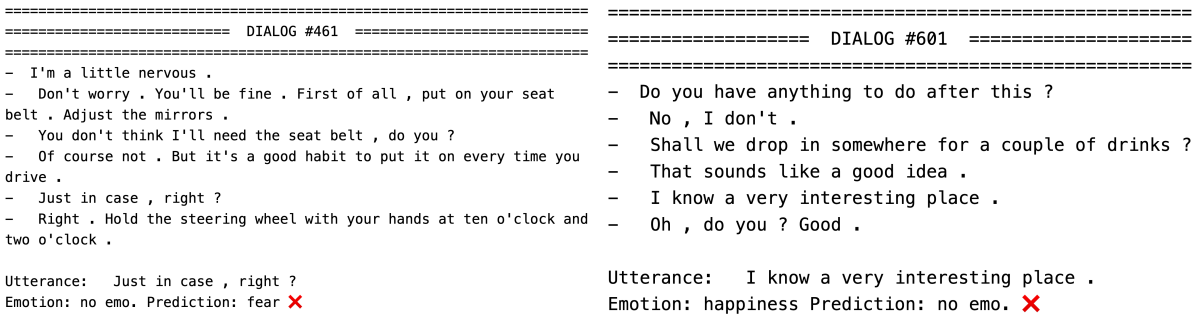


(a) (b)

Figure 4.11: Two examples of unexplainable predictions.

All in all, what emerges from the aforementioned points is that sometimes the context seems to be taken into account, sometimes the prediction is strongly based on the utterance itself. Therefore we definitely cannot state that context helps in all cases, which also means that we cannot assert it is accurately represented in such model. This may be the main limitation of our contextual model. More details on this point are provided in the next section.

However, there is an overall relevancy of predictions, as a majority of them are correct and also because the presence of an unknown feeling is often detected by the model. What can be added to that point is that some predictions said to be wrong can actually be considered as plausible or even true. This brings us back straight to the question of subjectivity in annotation, and leads to situations like those illustrated in Figure 4.12. In this case, we can reasonably consider the prediction as accurate for the utterance, even if it is not the expected label.



(a) (b)

Figure 4.12: Two examples of situation when the predicted labels, although false, seems to be a legitimate predictions.

5. Limitations

In this chapter, we take a step back from the results presented in the previous chapter to discuss them and identify biases or shortcomings in our approach. First, we discuss some general aspects, linked to the chosen dataset and the meta-learning algorithms as such. Next, we look more specifically at the representation of conversational context and the way we implemented it, which is the main objective and contribution of our new approach.

5.1 General Concerns

Data specificity. For now, we evaluate emotion detection only on DailyDialog corpus, which is actually a restrictive setup. Indeed, artificially generated conversations are not bound to accurately reflect human interactions. They should differ both regarding emotions distribution and the way these emotions are expressed. Thus, the proposed models should also be evaluated on real conversations to ensure their transferability to other dataset.

In addition, as we already mentioned, the dataset is highly unbalanced. This has been partially solved by using triplet loss and a weighted CE loss for the encoder of the conversational model, although it is not a completely solved issue. Anyway, a shortcoming remains with data distribution. As it can be seen in Figure 5.2, the data distribution regarding emotion labels is very similar between train and test set, which introduces significant bias. Therefore, we have to be cautious during the generalization step and ensure that the model will adapt correctly to other data distribution.

Overall performance. Eventually, the scores reached thanks to this approach are mainly quite low. We obtain a weighted F1 of 68% in isolated case and 51% in conversational case. It is possible that these results are due to the use of metric-based learning in general which might not be the most adapted approach to perform meta-learning (even if it theoretically consists in meta-learning). Thus, in order to broaden metric learning methods, it might be relevant to develop a model architecture that combines metric learning to another deep learning method, such as self-attention layers. Even though, there is a more general concern regarding meta-learning, whose approaches are known to provide lower performances that pre-trained models. Therefore, in terms of purely task resolution, meta-learning models are not bound to give the best results on the selected dataset, but should still allow acceptable results in specific situations (few data, unknown labels, *etc.*). This is what makes this approach still interesting in such context.

5.2 Conversation-Aware Representations Handling

Computational heaviness. It is actually quite long to run the context-aware model on the whole training set, and the reason for that should be that the training process in itself is actually not optimized. This is mostly due to the encoding part. Indeed, using a usual BERT encoder consists in a very heavy dialog representation, in addition to not being really adapted because of truncation. To solve both the issues of heaviness and undesired

truncation, a solution would be to use an encoder such as SentenceBERT [57] that provide a representation at the sentence-level instead of word-piece level. Then, in order to fully account for conversational context, one would add some transformer encoder layers to bring attention across utterances of the same dialog. This alternative approach should provide a more hierarchical representation of the knowledge conveyed by the utterances within a conversation.

Few-shot performances. This concern points that the Siamese Network architecture might not be the best in order to perform few-shot learning. This is a critical issue in the meta-learning setting where one is expected to observe significant generalization abilities across (potentially unseen) emotions. A more adapted approach for that would be MAML (Model-Agnostic Meta-Learning) which seems to be an appropriate method to acquire transferable knowledge from initialization [16], and which is actually the second part of the internship.

Conclusion and Perspectives

Project review. To start with, the main challenge was to develop a way to take into account the conversational context, which, at the best of our knowledge, makes of this work an actual contribution to state-of-the-art methods. Once we ended up with such conversation-aware representations, the next step has been to build an end-to-end deep learning structure that performs emotions detection using some meta-learning setting. More precisely, the selected approaches follow the meta-learner strategy to enable transferability of knowledge about emotions. It has been evaluated regarding quantitative and qualitative criteria in order to better understand the bias and challenges of emotion recognition in conversation. In particular we could learn that such model training is challenging, especially when we want to implement context-awareness. Despite this, when studied individually predictions seem encouraging in terms of retained knowledge. This work is therefore a very first step towards generalization through emotions.

Future work. At this stage, there is still work to be done as the project will actually last two more months at time of this writing. What appears to be the most relevant axis for future work at this point is to build a MAML architecture, as it proved efficient in such context. Thus, the Siamese Networks would serve as a baseline to start a basic benchmark on emotion detection in conversation. In addition, in order to ensure stable training for this second model, it will be necessary to provide some control on the learning rate, using either a decay rate or a learning rate scheduler.

Personal take-aways. This Master Thesis is a real opportunity to improve my technical skills in deep learning and computer science. Indeed, I needed to learn PyTorch deep learning framework from the very beginning in order to be able to design custom models. This is maybe the most challenging part of the Master Thesis I encountered so far, and I am gradually upskilling on this framework as I managed to develop a whole meta-learning model architecture along with preprocessing, training and evaluation. In addition to my main task, as part of SyNaLP team I had the occasion to take part in lab events and activities, starting with the French text mining challenge DEFT (Défi Fouille de Textes) [1]. Thanks to the support of my supervisor I was able to be part of his team working on this challenge which led to a publication [7]. This participation also gave me the chance to attend a joint French conference on NLP and Information Retrieval: CORIA-TALN 2023 [2].

All these experiences have been truly profitable for me because I plan to start a PhD by the end of the year. This immersive experience in research is still giving me many opportunities to interact with the NLP community and work on several state-of-the-art approaches. Therefore, I am very grateful that I can work with SyNaLP today and I am looking forward to pursuing work collaboration at the LORIA laboratory[3].

[1] `https://deft2023.univ-avignon.fr/` (website in French)
[2] `https://coria-taln-2023.sciencesconf.org/` (website in French)
[3] `https://www.loria.fr/en/`

Bibliography

[1] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*, 2(7):e12189, 2020.

[2] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations*, 2018.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015*, 1409, 09 2014.

[4] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics (Oxford, England)*, 16:412–24, 06 2000.

[5] D. Balouek, A. C. Amarie, G. Charrier, F. Desprez, E. Jeannot, E. Jeanvoine, A. Lèbre, D. Margery, N. Niclausse, L. Nussbaum, O. Richard, C. Perez, F. Quesnel, C. Rohr, and L. Sarzyniec. Adding virtualization capabilities to the grid'5000 testbed. In I. I. Ivanov, M. van Sinderen, F. Leymann, and T. Shan, editors, *Cloud Computing and Services Science*, pages 3–20, Cham, 2013. Springer International Publishing.

[6] C. Blanc, A. Bailly, Élie Francis, T. Guillotin, F. Jamal, B. Wakim, and P. Roy. Flaubert vs. camembert: Understanding patient's answers by a french medical chatbot. *Artificial Intelligence in Medicine*, 127:102264, 2022.

[7] A. Blivet, S. Degrutère, B. Gendron, A. Renault, C. Siouffi, V. G. Bouju, C. Cerisara, H. Flamein, G. Guibon, M. Labeau, and T. Rousseau. Participation de l'équipe ttgv à deft 2023 : Réponse automatique à des qcm issus d'examens en pharmacie. *Actes de 18e Conférence en Recherche d'Information et Applications, 16e Rencontres Jeunes Chercheurs en RI, 30e Conférence sur le Traitement Automatique des Langues Naturelles, 25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (CORIA-TALN'2023), Paris (France)*, 2023.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[9] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan. Iemocap: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359, 2008.

[10] A. Chatterjee, K. N. Narahari, M. Joshi, and P. Agrawal. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th Inter-*

national Workshop on Semantic Evaluation, pages 39–48, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[11] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways, 2022.

[12] H. Cramér. *Mathematical Methods of Statistics (PMS-9), Volume 9*. Princeton University Press, Princeton, 1946.

[13] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, June 2019. Association for Computational Linguistics.

[15] T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, jan 2019.

[16] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1126–1135. JMLR.org, 2017.

[17] J. R. Fontaine, K. R. Scherer, E. B. Roesch, and P. C. Ellsworth. The world of emotions is not two-dimensional. *Psychological science*, 18(12):1050–1057, 2007.

[18] G. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[19] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil. Forward and reverse gradient-based hyperparameter optimization. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1165–1173. PMLR, 06–11 Aug 2017.

[20] T. Fujioka, T. Homma, and K. Nagamatsu. Meta-learning for speech emotion recognition considering ambiguity of emotion labels. In *INTERSPEECH*, pages 2332–2336, 2020.

[21] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[22] T. Gentner, T. Neitzel, J. Schulze, and R. Buettner. A systamatic literature review of medical chatbot research from a behavior change perspective. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 735–740, 2020.

[23] E. Grant, C. Finn, S. Levine, T. Darrell, and T. L. Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[24] G. Guibon, M. Labeau, H. Flamein, L. Lefeuvre, and C. Clavel. Few-shot emotion recognition in conversation with sequential prototypical networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Punta Cana, Dominican Republic, 2021.

[25] G. Guibon, M. Labeau, H. Flamein, L. Lefeuvre, and C. Clavel. Meta-learning for classifying previously unseen data source into previously unseen emotional categories. In *Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing*, page 76–89, Online, Aug 2021. Association for Computational Linguistics.

[26] J. Herzig, G. Feigenblat, M. Shmueli-Scheuer, D. Konopnicki, A. Rafaeli, D. Altman, and D. Spivak. Classifying emotions in customer support dialogues in social media, 09 2016.

[27] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[28] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, Sep 2022.

[29] C.-C. Hsu, S.-Y. Chen, C.-C. Kuo, T.-H. Huang, and L.-W. Ku. EmotionLines: An emotion corpus of multi-party conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[30] D. Hu, L. Wei, and X. Huai. DialogueCRN: Contextual reasoning networks for emotion recognition in conversations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7042–7052, Online, Aug. 2021. Association for Computational Linguistics.

[31] M. I. Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986, 5 1986.

[32] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition, 2015.

[33] J. D. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

[34] P. H. Le-Khac, G. Healy, and A. F. Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[35] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan, Nov. 2017. Asian Federation of Natural Language Processing.

[36] Z. Li, F. Tang, M. Zhao, and Y. Zhu. EmoCaps: Emotion capsule based model for conversational emotion recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1610–1618, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[37] C. Liang, J. Xu, Y. Lin, C. Yang, and Y. Wang. S+PAGE: A speaker and position-aware graph neural network model for emotion recognition in conversation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 148–157, Online only, Nov. 2022. Association for Computational Linguistics.

[38] X. Lin, H. Baweja, G. Kantor, and D. Held. Adaptive auxiliary task weighting for reinforcement learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[39] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search, 2019.

[40] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[41] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, and E. Cambria. Dialoguernn: An attentive rnn for emotion detection in conversations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6818–6825, Jul. 2019.

[42] B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et biophysica acta*, 405 2:442–51, 1975.

[43] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner, Feb 2018. arXiv:1707.03141 [cs, stat].

[44] B. Mor, S. Garhwal, and A. Loura. A systematic review of hidden markov models and their applications. *Archives of Computational Methods in Engineering*, 28, 05 2020.

[45] A. Neuraz, L. Campillos Llanos, A. Burgun, and S. Rosset. Natural language understanding for task oriented dialog in the biomedical domain in a low ressources context, nips workshop. In *Machine Learning for Health (ML4H): Moving beyond supervised learning in healthcare*, Montréal, Québec, Canada, 2018.

[46] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural networks : the official journal of the International Neural Network Society*, 113:54–71, 2018.

[47] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[49] K. Pearson. Vii. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, 58(347-352):240–242, 1895.

[50] P. Pereira, H. Moniz, and J. P. Carvalho. Deep emotion recognition in textual conversations: A survey, Nov 2022. arXiv:2211.09172 [cs].

[51] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 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, June 2018. Association for Computational Linguistics.

[52] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy, July 2019. Association for Computational Linguistics.

[53] S. Poria, N. Majumder, R. Mihalcea, and E. Hovy. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953, 2019.

[54] W. Ragheb, J. Azé, S. Bringay, and M. Servajean. Attention-based modeling for emotion detection and classification in textual conversations. *CoRR*, abs/1906.07020, 2019.

[55] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy, July 2019. Association for Computational Linguistics.

[56] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, Nov 2016.

[57] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[58] S. Ritter, J. X. Wang, Z. Kurth-Nelson, S. M. Jayakumar, C. Blundell, R. Pascanu, and M. Botvinick. Been there, done that: Meta-learning with episodic recall, 2018.

[59] S. Ruder. An overview of gradient descent optimization algorithms, 2017.

[60] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[61] J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[62] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4080–4090, Red Hook, NY, USA, 2017. Curran Associates Inc.

[63] X. Song, L. Huang, H. Xue, and S. Hu. Supervised prototypical contrastive learning for emotion recognition in conversation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5197–5206, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[64] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[65] M. Tkalcic, A. Kosir, and J. Tasic. Affective recommender systems: the role of emotions in recommender systems. In *Proc. The RecSys 2011 Workshop on Human Decision Making in Recommender Systems*, pages 9–13, 2011.

[66] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[67] N. Van der Heijden, S. Abnar, and E. Shutova. A comparison of architectures and pretraining methods for contextualized multilingual word embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:9090–9097, 04 2020.

[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[69] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3637–3645, Red Hook, NY, USA, 2016. Curran Associates Inc.

[70] L. Xu, L. Sanders, K. Li, and J. Chow. Chatbot for health care and oncology applications using artificial intelligence and machine learning: Systematic review. *JMIR Cancer*, 7, 2021.

[71] T. Yang, X. Yu, N. Ma, Y. Zhang, and H. Li. Deep representation-based transfer learning for deep neural networks. *Knowledge-Based Systems*, 253:109526, 2022.

[72] Y. Yang and T. M. Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *International Conference on Learning Representations*, 2017.

[73] S. M. Zahiri and J. D. Choi. Emotion detection on tv show transcripts with sequence-based convolutional neural networks, 2017.

Appendix

Figure 5.1: Distribution of expressed emotions *w.r.t.* the utterance index in dialog, for each emotion.

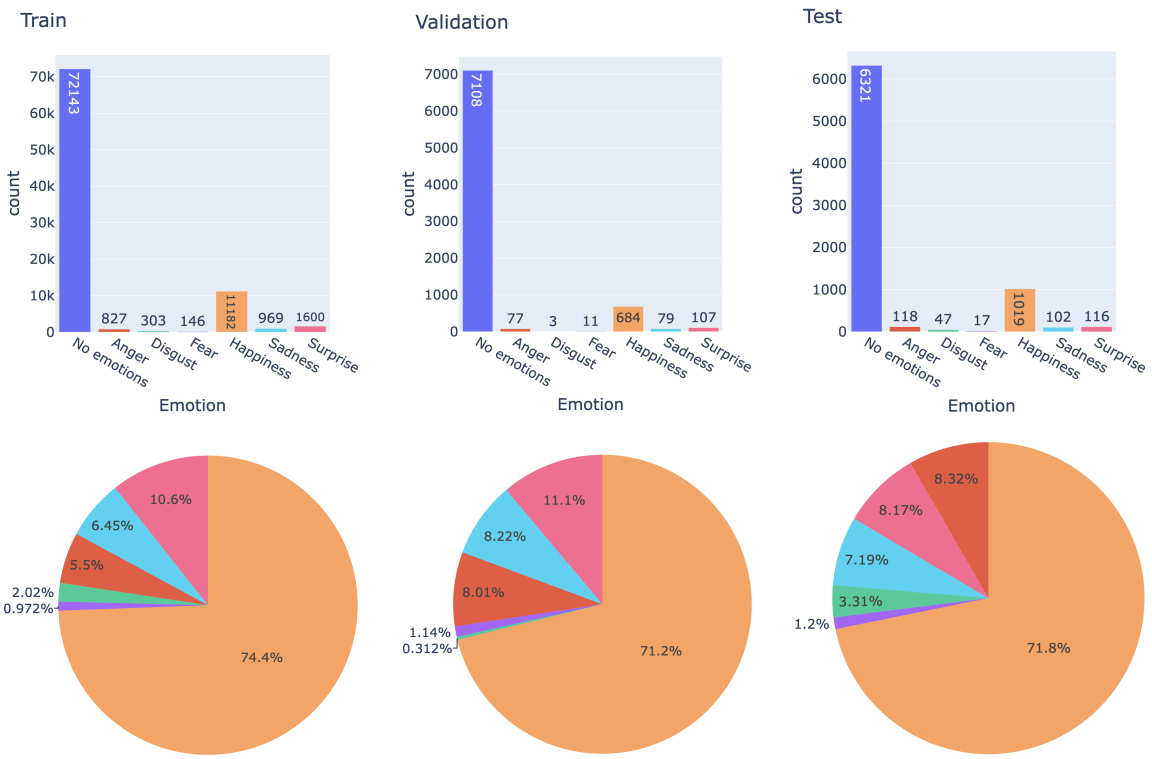


Figure 5.2: Distribution of emotions in each DailyDialog split. Each column gives the barplot for all labels and the pie chart for emotional labels only.

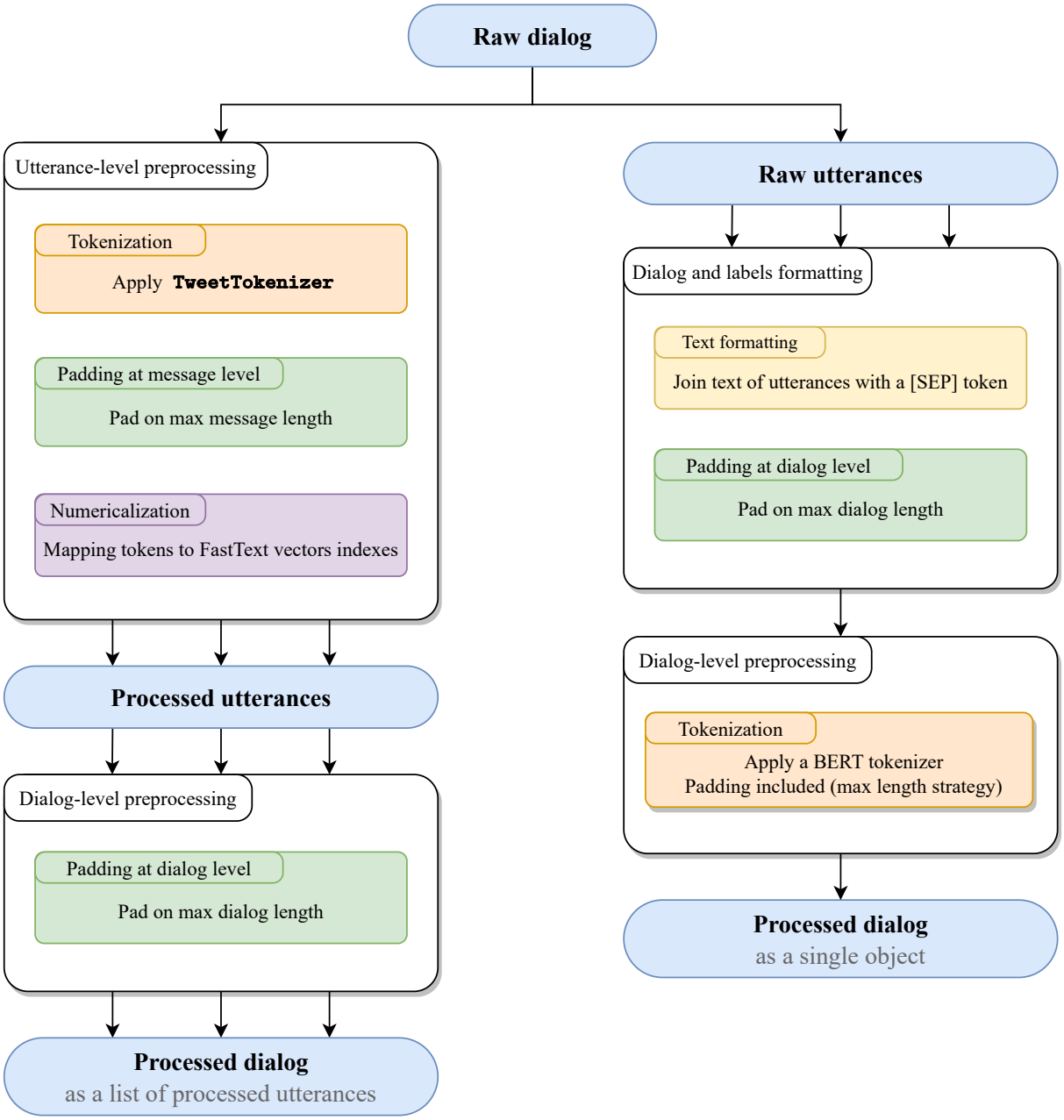
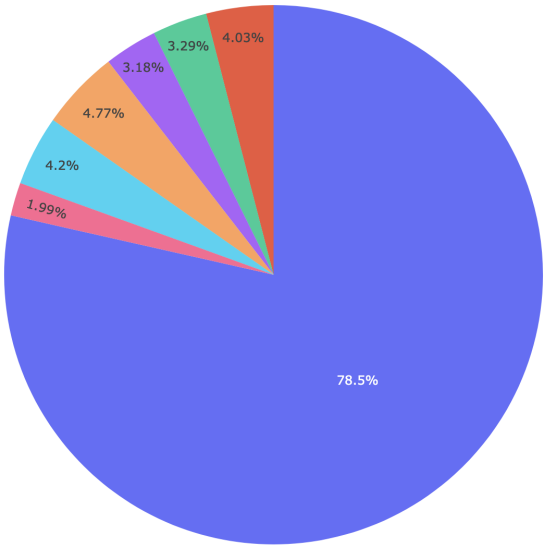


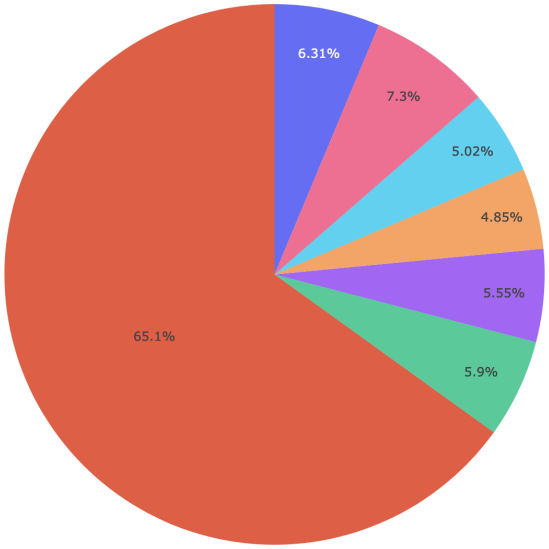
Figure 5.3: The detailed pre-processing pipeline to obtain individual utterance representations (left) or conversation-aware dialog representations (right).

```
## Original dialog

- Say, Jim, how about going for a few beers after dinner ?
- You know that is tempting but is really not good for our fitness.
- What do you mean ? It will help us to relax.
- Do you really think so ? I don't. It will just make us fat and act silly. Remember last time ?
- I guess you are right. But what shall we do? I don't feel like sitting at home.
- I suggest a walk over to the gym where we can play singsong and meet some of our friends.
- That's a good idea. I hear Mary and Sally often go there to play pingpong. Perhaps we can make a foursome
with them.
- Sounds great to me! If they are willing, we could ask them to go dancing with us. That is excellent
exercise and fun, too.
- Good. Let's go now.
- All right.

## Formatted dialog with [SEP] tokens

Say , Jim , how about going for a few beers after dinner ? [SEP] You know that is tempting but is really not
good for our fitness . [SEP] What do you mean ? It will help us to relax . [SEP] Do you really think so ? I
don't . It will just make us fat and act silly . Remember last time ? [SEP] I guess you are right.But what
shall we do ? I don't feel like sitting at home . [SEP] I suggest a walk over to the gym where we can play
singsong and meet some of our friends . [SEP] That's a good idea . I hear Mary and Sally often go there to
play pingpong.Perhaps we can make a foursome with them . [SEP] Sounds great to me ! If they are willing , we
could ask them to go dancing with us.That is excellent exercise and fun , too . [SEP] Good.Let ' s go now .
[SEP] All right .

## Word-piece tokenization

['[CLS]', 'say', ',', 'jim', ',', 'how', 'about', 'going', 'for', 'a', 'few', 'beers', 'after', 'dinner', '?
', '[SEP]', 'you', 'know', 'that', 'is', 'tempting', 'but', 'is', 'really', 'not', 'good', 'for', 'our',
'fitness', '.', '[SEP]', 'what', 'do', 'you', 'mean', '?', 'it', 'will', 'help', 'us', 'to', 'relax', '.', '
[SEP]', 'do', 'you', 'really', 'think', 'so', '?', 'i', 'don', "'", 't', '.', 'it', 'will', 'just', 'make',
'us', 'fat', 'and', 'act', 'silly', '.', 'remember', 'last', 'time', '?', '[SEP]', 'i', 'guess', 'you',
'are', 'right', '.', 'but', 'what', 'shall', 'we', 'do', '?', 'i', 'don', "'", 't', 'feel', 'like',
'sitting', 'at', 'home', '.', '[SEP]', 'i', 'suggest', 'a', 'walk', 'over', 'to', 'the', 'gym', 'where',
'we', 'can', 'play', 'sings', '##ong', 'and', 'meet', 'some', 'of', 'our', 'friends', '.', '[SEP]', 'that',
"'", 's', 'a', 'good', 'idea', '.', 'i', 'hear', 'mary', 'and', 'sally', 'often', 'go', 'there', 'to',
'play', 'ping', '##pon', '##g', '.', 'perhaps', 'we', 'can', 'make', 'a', 'fours', '##ome', 'with', 'them', '.
', '[SEP]', 'sounds', 'great', 'to', 'me', '!', 'if', 'they', 'are', 'willing', ',', 'we', 'could', 'ask',
'them', 'to', 'go', 'dancing', 'with', 'us', '.', 'that', 'is', 'excellent', 'exercise', 'and', 'fun', ',',
'too', '.', '[SEP]', 'good', '.', 'let', "'", 's', 'go', 'now', '.', '[SEP]', 'all', 'right', '.', '[SEP]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '
[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]', '[PAD]']

## Numericalization

[101, 2360, 1010, 3958, 1010, 2129, 2055, 2183, 2005, 1037, 2261, 18007, 2044, 4596, 1029, 102, 2017, 2113,
2008, 2003, 23421, 2021, 2003, 2428, 2025, 2204, 2005, 2256, 10516, 1012, 102, 2054, 2079, 2017, 2812, 1029,
2009, 2097, 2393, 2149, 2000, 9483, 1012, 102, 2079, 2017, 2428, 2228, 2061, 1029, 1045, 2123, 1005, 1056,
1012, 2009, 2097, 2074, 2191, 2149, 6638, 1998, 2552, 10021, 1012, 3342, 2197, 2051, 1029, 102, 1045, 3984,
2017, 2024, 2157, 1012, 2021, 2054, 4618, 2057, 2079, 1029, 1045, 2123, 1005, 1056, 2514, 2066, 3564, 2012,
2188, 1012, 102, 1045, 6592, 1037, 3328, 2058, 2000, 1996, 9726, 2073, 2057, 2064, 2377, 10955, 5063, 1998,
3113, 2070, 1997, 2256, 2814, 1012, 102, 2008, 1005, 1055, 1037, 2204, 2801, 1012, 1045, 2963, 2984, 1998,
8836, 2411, 2175, 2045, 2000, 2377, 17852, 26029, 2290, 1012, 3383, 2057, 2064, 2191, 1037, 23817, 8462,
2007, 2068, 1012, 102, 4165, 2307, 2000, 2033, 999, 2065, 2027, 2024, 5627, 1010, 2057, 2071, 3198, 2068,
2000, 2175, 5613, 2007, 2149, 1012, 2008, 2003, 6581, 6912, 1998, 4569, 1010, 2205, 1012, 102, 2204, 1012,
2292, 1005, 1055, 2175, 2085, 1012, 102, 2035, 2157, 1012, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

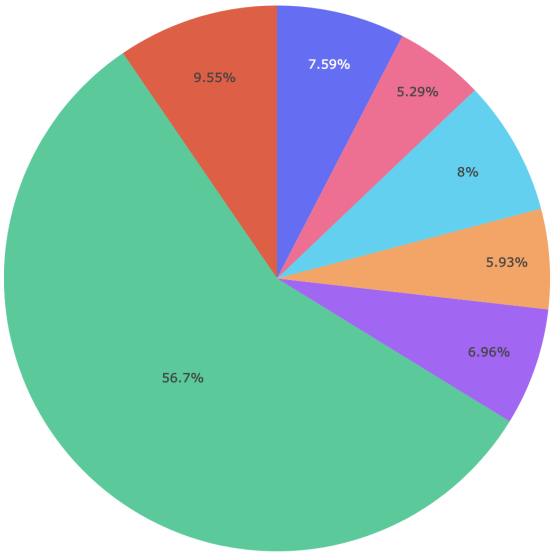
Figure 5.4: An example of application of the preprocessing pipeline to apply a BERT model.



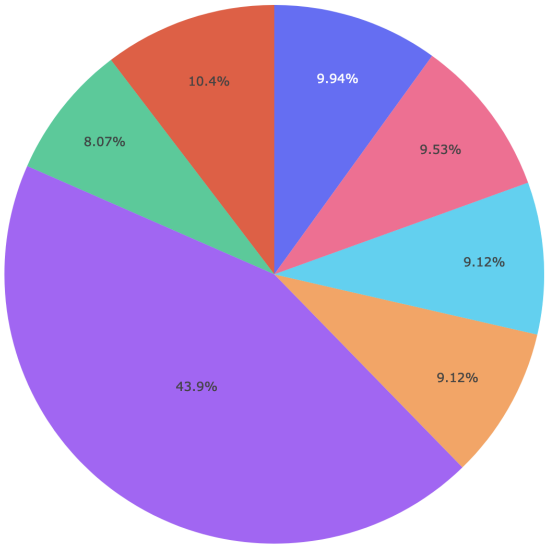
No emotion



Anger



Disgust



Fear

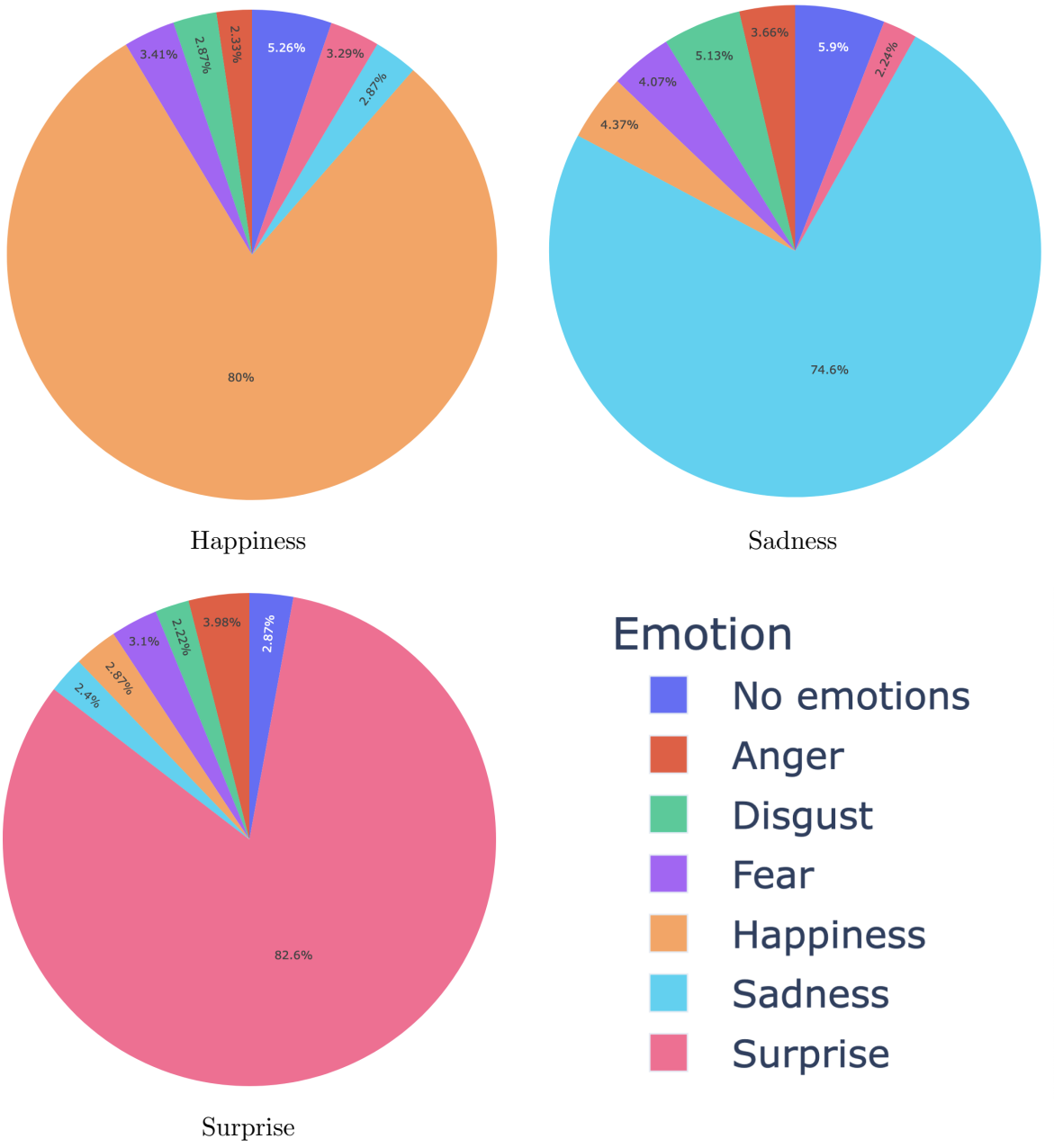
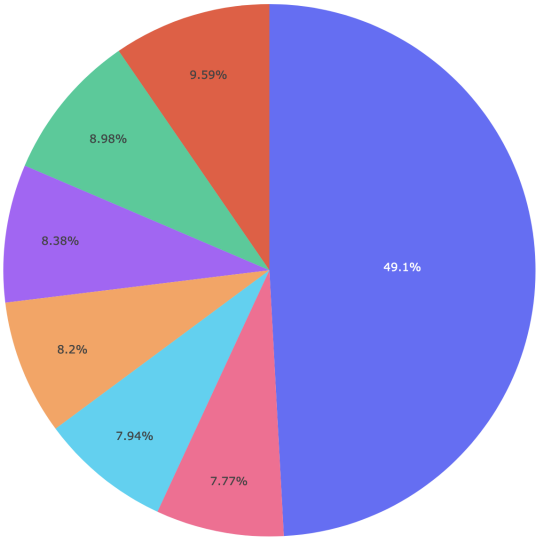
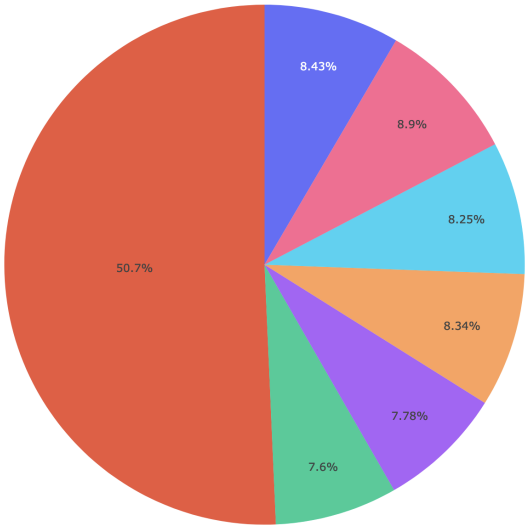


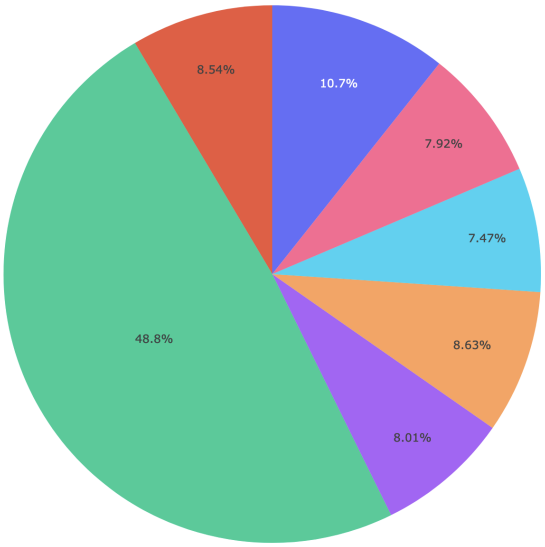
Figure 5.6: Distribution of predictions for each actual emotion in the case of isolated utterances representations.



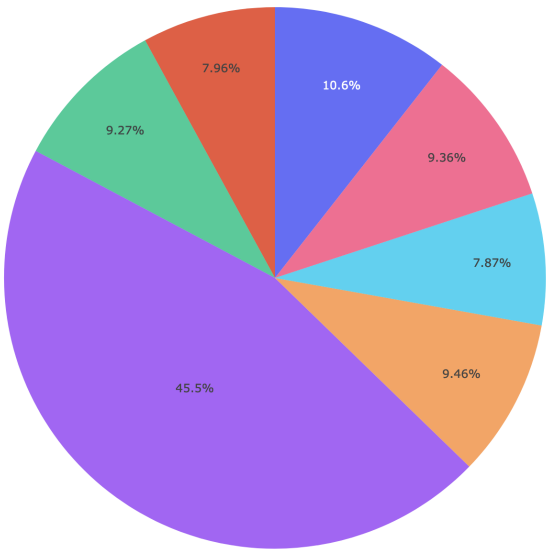
No emotion



Anger



Disgust



Fear

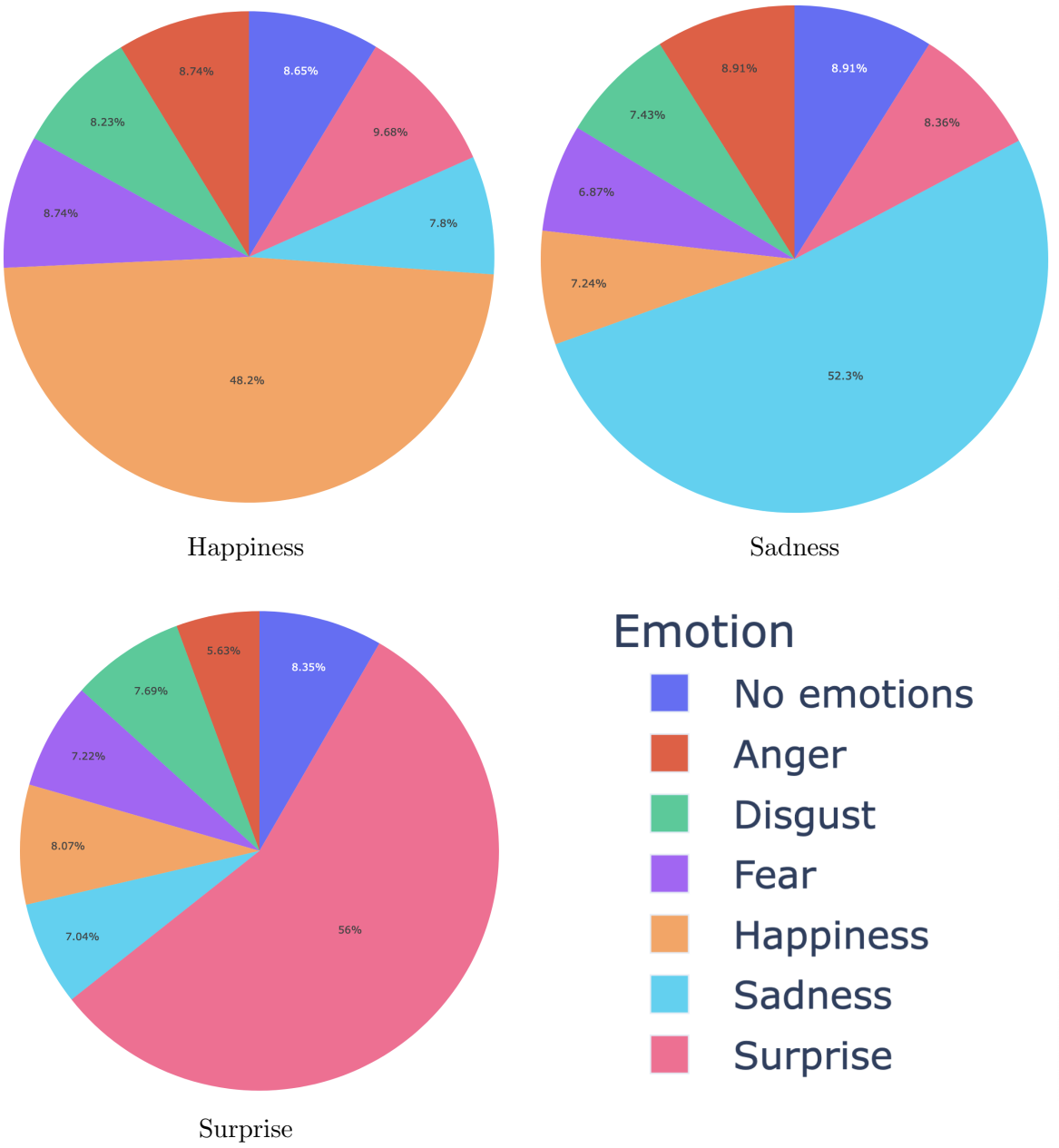


Figure 5.8: Distribution of predictions for each actual emotion in the case of isolated utterances representations.