Msc Natural Language Processing 2023-2024

SOFTWARE PROJECT

Which Approach to Detect Irony?

Students:
Pierre EPRON
Maxime RENARD
Shu ZHANG

Supervisors:
Miguel COUCEIRO
Gaël GUIBON

February 5, 2024

1 Introduction

Irony is a complex linguistic phenomenon with varying interpretations that poses a challenge for both humans and automated systems. Recognizing irony is crucial, especially in the context of harmful behavior on social media [1, 2].

Leveraging recent advancements and a rich irony dataset, this project aims to enhance irony detection using state-of-the-art language models, comparing the performances of traditional Language Models against the performances of Large Language Models (LLMs), using two irony datasets, one focused on the different perspectives of the annotators and the other focused on the different types of irony.

1.1 What is irony?

Most theories on understanding irony involve violating the norms of cooperative communication. In other words, to understand an ironic proposition, you have to understand the opposite of its literal meaning. This principle is also known as "violating the maxim of quality" [3].

Most theories also define two main types of irony: **Verbal** and **Situational** Irony. Verbal irony is a type of irony where a person explicitly says something to express the opposite, expressing humor, frustration, anger or other moods. Situational irony represents ironic propositions or scenarios that deviate from anticipated outcomes. As illustrated in [4], a classic example of situational irony involves firefighters experiencing a fire in their own kitchen while responding to a fire alarm.

Two more categories have been formalized by scholars:

Dramatic or Tragic irony [5], which is a specific case of situational irony implying that the audience knows something that the characters do not. A classic example of situational irony is the story of Oedipus. Oedipus didn't know that he was adopted, killed his biological father and slept with his biological mother.

Discourse irony [6], which claims that "all ironic comments have the necessary characteristic of allusion to a discrepancy between what occurs and what should have occurred." with the given example of someone saying "How about another small slice of pizza?" to someone who just gobbled up the whole pie.

Researchers in psycho-linguistics have looked into the thought processes underlying irony. The work of [7] summarises a number of findings on the mechanisms for understanding irony. They have shown that understanding the meaning of statements that use irony does not necessarily depend on recognizing the irony itself. Furthermore, the identification of ironic meanings is not linked to any prosody feature such as intonation. As a result, irony can be understood with text alone.

The results of their experiments run counter to the main theory of the maxim of quality. Understanding irony is not always linked to a violation of this maxim. It is often more related to a memory of a past statement or shared beliefs. These findings reinforce an existing theory: the echoic mention theory [8, 9]. Following this theory, we understand the irony when the listener is reminded by an echo of a familiar proposition and by the speaker's attitude towards it. The common example used to describe it is about a mother who says to her son "I love children who keep their rooms clean" to tell her son to tidy his room. The irony comes from the agreements made between them to keep the room clean.

Finally, they showed that irony in a proposition is not always intended by the speaker. They asked a group of participants to read two variants of the same story (18 pairs of stories in total), the first with the speaker voluntarily expressing the ironic proposition and the second with the speaker involuntarily expressing the ironic proposition. They found that, on average, readers spent less time reading the unintentional proposition (2.33 seconds compared with 3.14) and judged it to be more ironic (5.2/7 compared with 4.3/7). This suggests that the irony of a proposition is partly independent of the speaker's will and even that we tend to find a speaker's propositions more ironic if they are not aware that they are.

The conclusions of these studies highlight the difficulties inherent in annotation and therefore in irony detection. These issues are summarized below.

1.2 Why is irony difficult to annotate and predict?

Differences: The understanding of irony is often affected by language and culture. Examples and situations mentioned in the text may have different ironic meanings in different cultures and contexts. This difference increases the difficulty of annotation and predicting irony. **Speaker intention**: Sometimes, irony may not be the speaker's explicit intention. Annotators need to understand irony while taking into account possible implicit intentions and the possibility of multiple interpretations. It increases the difficulty of annotating and predicting irony.

Context dependence: Irony usually relies on a specific context to produce its effect. The same irony may have different effects in different contexts. If context dependence is considered, annotating irony will be a time-consuming project. This is also a significant challenge for annotation and prediction.

Creativity and Flexibility: Irony is usually a creative form of expression that is flexible and imaginative. Therefore, irony often has unexpected content and form, which is difficult to predict using conventional methods and logic.

1.3 Motivations

Studies have shown that LLMs and their ability to encode knowledge [10] offer new perspectives for solving complex tasks such as irony detection. Other works also showed that LLMs can be efficient in an unsupervised approach [11]. However, complicated tasks require the need to fine-tune the model which is often costly in terms of time and hardware. Refined learning techniques like the use of Low-rank adapter (LoRA) [12] have offered suitable solutions to address the problem. For the moment, the ability of LLMs to detect irony has not been extensively tested. The main objective of this paper is therefore to compare the performances of traditional Language Models against the performances of LLMs and their different learning approaches. We argue that their knowledge embeddings could lead to significant improvement in the field of Irony detection.

2 Related work

The work[1] combines linguistic and meta-linguistic features, it is assumed that these features indicate the ironic content in the text. Here are the features considered in the model:

Polarity: This evaluates whether words in the text have a positive or negative connotation. For example, "What a beautiful view," said about a wall without windows. (Positive words used in a context where the literal interpretation would be negative.)

Unexpectedness: A measure of the extent to which a result or statement in a text is surprising or deviates from the norm. For example, "Great job on cleaning your room," when the room is messier than before. (The statement is unexpected given the context.)

Emotional scenarios: This feature captures the emotional context or mood expressed by the text. Ironic statements often carry emotional implications that contradict the literal meaning. For example, "I'm absolutely thrilled to be working overtime again this weekend," spoken by an employee who is frustrated. (The emotional scenario conveyed by "thrilled" is opposite to the likely true feelings.)

Ambiguity: Irony can depend on ambiguity at structural, morphosyntactic and semantic layers, exploiting words or situations that can be interpreted in multiple ways. For example, "Oh, I just love paying taxes," when said ironically. (The statement can be interpreted as genuine or ironic, thus ambiguous.)

The combination of these features led to a marked improvement in the accuracy of classifiers. While individual features perform well, their combination significantly improves performance, achieving a maximum accuracy of 91.97% when distinguishing satirical content from political content.

2.1 TweetEval

One of the most popular corpora is SemEval-2018 [13]. It is part of a larger project which consisted of collecting tweets and annotating them using several modalities such as emojis, sentiments, etc. And including irony. They started collecting 3,000 tweets, using hashtags in tweets to have an initial automatic classification. Thus, tweets mentioning #irony, #sarcasm and #not were considered as ironic. This presupposes that the irony of a proposition is always intended by its authors, which as explained previously, is not always the case.

They then performed manual annotation. The annotation was made by 3 non-native English speaker linguistics students. Their first goal was to correct the automatic annotation made during the data collection process. They also introduced a new modality to specify the type of irony encountered. They used four labels: **verbal irony by polarity contrast**, **other verbal irony**, **situational irony** and **non-irony**.

The first label, **verbal irony by polarity contrast**, is defined as a proposition where the understood meaning is the opposite of the literal meaning, whether positive or negative. Overall, this is the type of verbal irony that satisfies the violation of the maxim of quality. Example: "Pretty excited about how you gave up on me".

The second label, **other verbal irony**, is defined as verbal irony that does not contain a polarity contrast. Example: "I just wrote an 8 pages paper... I was awfully tired when I was writing it and now I can't sleep."

The third label, **situational irony**, covers all kinds of situational irony propositions. Example: "If you wanna look like a badass, have drama on social media".

The fourth label is used for non ironic propositions.

They used a standard procedure to assess the consistency of their annotations. They sampled 100 tweets from their corpus and performed two rounds of annotations, which they evaluated by calculating Fleiss' Kappa score. Between the two rounds, they debated their different results.

Fleiss' kappa (κ) ([14]) is a statistical measure used to evaluate the agreement between several annotators when they annotate items in several classes. It is an extension of Cohen's kappa, which is used in the context of a binary modality. Like Cohen' κ score, Fleiss' κ score is specifically designed for contexts where there are more than two annotators. This type of score is generally referred to as the inter-annotator agreement score (IAA score). Both scores are bounded between 0 and 1, 0 being a total disagreement and 1 being a total agreement.

For the ironic/not ironic modality, they report an IAA score of 0.65 after the first round and an IAA score of 0.72 (+0.07) after the second round. For the four classes modality, they report an IAA score of 0.55 after the first round and an IAA score of 0.72 (+0.17) after the second round. In both cases and taking into account the subjectivity of the task, this is an interesting result. But the scores still remain quite low and this shows the difficulty of agreeing on the perception of irony. We can also observe that the score on the 4-class modality improves much more than the score on the binary modality after the second round. This tends to imply that the perception of the irony of a proposition has not changed much after the debates of the first round but that the perception of the type of irony has become homogenized.

Semeval-2018 allowed the creation of one of the most coherent corpora in terms of detecting irony, particularly verbal irony. Yet they show that this is a subjective and complex task, which reinforces the fact that even if we share a common definition of what irony is, we still have different perceptions of what is ironic.

2.2 EPIC

The English Perspectivist Irony Corpus (EPIC) ([15]) is a disaggregated English corpus for irony detection, containing 3,000 pairs of short conversations (Post-Reply) from two social media sites, Twitter[1] and Reddit[2], along with the demographic information of each annotator (age, sex, ethnicity, country of birth, country of residence, nationality, whether they are a student or not and their employment status), with the stated goal of studying irony, and irony detection, from a Perspectivist point of view.

Reddit comments were collected from subreddits frequented by English-speaking users between January 2020 and June 2021, with subsequent processing to ensure English language and data integrity by removing irrelevant pairs and identifying language by using the LangID Python library[3]. The Twitter data collection used geolocation through the Twitter API to discern English varieties by validating the country associated with the tweet pairs. Queries to the Twitter Stream API were made to gather English tweets from the specified five countries, with a focus on *conversation starting* tweets and excluding replies or quotes.

In this study, each pair of Post-Reply was classified by multiple annotators tasked with assigning a binary label to the Reply, distinguishing between **Ironic** or **Not Ironic**, based on the information provided by the context in the Post. Seventy-six annotators, all native English speakers, were recruited by the researchers,

[1] twitter.com
[2] reddit.com
[3] LangID library

with each annotator responsible for evaluating 200 instances from the dataset. To ensure a comprehensive and diverse perspective on irony perception, instances were evenly distributed among annotators from five distinct English-speaking countries (United Kingdom, Ireland, United States, Australia and India). This deliberate approach yielded a final set of 74 validated annotators, resulting in a comprehensive dataset with a total of 14,172 annotations.

In 66% of the instances, at least one annotator disagreed with the rest. This leads to the main point of the article, regarding the importance of perspectivism when working with irony. Using Cohen's κ agreement score shows that selecting a label with the Majority Decision will ignore a group. For example, when their annotations were compared to the ones selected through Majority Decision, Asian annotators had a lower agreement than their White counterparts (average agreement of 0.414 and 0.493 respectively), which was then shown to be statistically significant. Another point brought up in the article is the variation in agreement based on the topic of the Post-Reply, where topics such as Labour or Science (average agreement of 0.6 and 0.575) were often more polarizing than topics such as health or arts (average agreement of 0.478 and 0.459)[4].

Finally, the researchers review the impact of factors such as sex, generation, and nationality on the perception of irony. Particularly noteworthy are the pronounced differences in perspective observed across generations.

This study stresses the importance of considering diverse viewpoints when developing models, even in challenging tasks like irony recognition. The identified disparities emphasize the need for more nuanced and inclusive approaches to ensure the creation of models that are both confident and representative of the diverse ways in which individuals interpret and understand irony.

Not included in the research article is an examination of the variance in agreement based on the source website of the Post-Reply pair. Although we were unable to find in-depth research on irony, [16] studies the evolution of language for people belonging to a web community. They found clues about language developments linked to prolonged (several months) use of web network (breastcancer.org in their case). We believe that there exists distinct *user cultures* between websites and in the case of EPIC, Reddit and Twitter. These differences likely stem from the purpose or format of the sites. Reddit functions as a social aggregation and discussion website, whereas Twitter serves as a microblogging social network. It is expected that they will attract different types of users and that even in the case of an overlap, users will behave and express themselves differently on the two platforms. Using the EPIC corpus, we applied Cohen's κ agreement score to compute the average agreement on pairs from Reddit and Twitter. This analysis reveals a significantly higher agreement (0.536 and 0.463 respectively) on instances from Reddit compared to Twitter, implying that users had an easier time classifying content from Reddit and underscoring the potential influence of platform-specific user culture on the interpretation of irony in online interactions.

2.3 Activator approach

[17] and [18] work focuses on annotating tweets using activators. They postulate that most ironic propositions contain two activators. These two activators share a relationship and this relationship allows irony to be perceived. Most of the time both activators are lexicalized (word phrases) but it may also happens that only one activator is lexicalized. This kind of proposition is considered as situational irony.

To achieve this goal, [18] sample tweets from the TWITTIRÒ-UD corpus[5] which is a Universal Dependency (UD) version of Italian tweets. Universal Dependencies (UD) is a framework for annotating grammatical structure in natural language. It aims to provide a cross-linguistically consistent representation that allows easy comparison of the syntax of different languages. The different linguistic features covered by UD are: Tokenization, Part-of-Speech (POS) and Dependency Relations. Using the dependency relation layer, they annotated segments of sentences as irony activators.

They carried out the annotation in 4 steps. The first step was to classify tweets as ironic, non-ironic, or undecided. They only kept the ironic tweets and moved on to the second step. This involved classifying ironic tweets as explicit or implicit. As explained above, the irony of a tweet is most often expressed by the presence of one or two lexicalized activators (P1, P2). Tweets containing two activators were considered explicit and tweets containing one activator were considered implicit (referring to an external activator). In this example, "(I love)P1 when my phone (turns the volume down automatically)P2", there are two explicit activators that express a contradiction and violate the maxim of quality. In this other example, "(#Hollande is really a good

[4]In this case, the agreement score varies between 1 and 0. A high agreement score means higher disagreement among annotators while a lower score indicates a high agreement.

[5]TWITTIRÒ-UD corpus

diplomat #Algeria)P1", there is only one explicit activator that contradicts an implicit activator that can be found in events that occurred before the tweet.

In steps 3 and 4, they refined their annotation by adding a subcategory of irony and clues. However, they do not report any measurement on this aspect and explain that for the sake of demonstration, they focus on Oxymoron and Paradox subcategories of irony. These subcategories refer to an explicit contradiction between the two activators. This is why they chose to concentrate on their work. This is surely the subcategory that is defined most simply and clearly possible. Example: "(Clubbing and putting up eyes)P1, (is not violent and it does obey the laws)P2."

They report an IAA for the activators' annotation on Oxymoron and Paradox irony. This IAA was computed on the annotation made by 2 skilled annotators on 277 tweets. They mentioned training on 50 tweets without more details. They computed 3 different scores. Total agreement, the proportion of tweets where annotators agreed on both activators, 40.9%. Partial agreement, the proportion of tweets where annotators agreed on one activator (P1 or P2), 34.1%. Disagreement, the proportion of tweets where annotators did not agree on both activators, 25.1%. These results show that focusing on a specific type of irony (Oxymoron and Paradox) and a specific characteristic of irony (activators) is still not sufficient to provide a strong and robust irony annotation. Their work remains interesting and introduces a new approach, based on intermediate features to annotate irony. They also provide complete and comprehensive guidelines on their annotation process, which is a valuable material for anyone attempting this exercise.

3 Method

3.1 RoBERTa

To perform the comparison with LLMs, we fine-tuned a RoBERTa model. RoBERTa [19](Robustly optimized BERT approach) is a variant of the popular BERT [20] (Bidirectional Encoder Representations from Transformers) model. Developed by Facebook AI, RoBERTa is trained according to the Masking Language Modeling (MLM) task. This consists of predicting one or more missing tokens in the sentence based on the context provided by the other words in the sentence. It is done by masking a random sample of tokens in the input sentence and training the model to predict this sample based on the remaining tokens. RoBERTa is designed to enhance the pretraining of natural language understanding by addressing certain limitations of BERT. It removes the next sentence prediction objective, utilizes larger mini-batches and learning rates, and trains on more extensive datasets. This approach leads to improved model performance across a wide range of natural language processing tasks, making RoBERTa a robust and versatile language representation model.

The version used in this paper "cardiffnlp/twitter-roberta-large-2022-154m" [21] responds to the different tasks proposed by TweetEval. To do this, it was trained on 154 million tweets sampled between January 2018 and December 2022. In this article, the EPIC dataset is extremely imbalanced, we tested Matthew Correlation Coefficient (MCC) Loss against a classic Cross-Entropy loss. The goal was to see if it could work better or converge faster at least.

3.2 Matthew Correlation Coefficient(MCC)

The MCC is a metric commonly used in binary classification tasks to evaluate the performance of a machine learning model. It takes into account true positives, true negatives, false positives, and false negatives, providing a balanced measure even when there is an imbalance between the classes. The MCC ranges from -1 to 1, where 1 indicates perfect prediction, 0 suggests random performance, and -1 implies complete disagreement between the model's predictions and the actual outcomes. A higher MCC value indicates better overall performance, considering both sensitivity and specificity, making it a robust choice for assessing binary classification models.

MCC has been introduced as a loss function for train image binary segmentation models [22]. They experimented with the loss on a skin lesion segmentation. The task consists of predicting a mask of black and white pixels which respectively represent negative and positive predictions. They define MCC loss as follows:

$$MCC = \frac{(TP.TN) - (FP.FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

$$MCCLoss = 1 - MCC$$

Where TP, TN, FP, FN stand for True Positive, True Negative, False Positive and False Negative, they are easy to compute given prediction values. As the Roberta architecture is designed to output two logits. We implement MCCLoss as follows:

$$Loss = \frac{MCCLoss(y, 1 - \hat{y}_0) + MCCLoss(y, \hat{y}_1)}{2}$$

Where $\hat{y}_0 \in (0, 1)^n$ is a vector of logits for negative predictions, $\hat{y}_1 \in (0, 1)^n$ is a vector of logits for positive predictions and $y \in [0, 1]^n$ is a vector of gold values.

3.3 Llama

Llama2[23], released by MetaAI under the Llama 2 Community License Agreement[6], is one of the most recent collections of LLM. It was pretrained on a corpus of 2 trillion tokens with a context size of 4,096. Llama2 is trained according to the Causal Language Modeling (CLM) task. This consists of predicting a token of a sentence based on the context provided by the previous tokens of the same sentence. The main difference with MLM is that CLM does not use all the context to predict a token but only the left context. It makes this kind of model more adapted to text generation. The collections contain models with different parameter size: 7 billion, 13 billion and 70 billion. It also contains fine-tuned versions of each model for the chat application. In this paper, we used Llama-2-7b-chat-hf[7] for zero-shot experiments and Llama-2-7b-hf[8] for LoRA experiments.

Using a prompt to encode a specific task, CLM can also be used as a base model for solving many different tasks using text generation. The work [24] gives a good overview on how to use CLM models and prompts to perform classification. There are many different approaches and in this paper, we will focus on two of them described, in the following sections.

3.3.1 Open and Close answer

These two approaches consist of using a fine-tuned LLM on the chat application to classify a given input. The main difference between the two methods is the decoding phase. In the open approach, we let the model generate the answer by using a beam-search decoder. In the close approach, we provide a partial answer like "The answer is " to the model and predict only the last token on the answer. By using a softmax activation function on the logits of the token labels, we can predict the most probable label. In our case of irony detection, we have two labels: "yes" and "no". Examples of prompts for the two approaches can be found in appendix A.1.

The advantage of the close approach is that it does not require any post-processing. The prediction is given by an argmax on the results of the softmax activation. In an open approach, the prediction should be inferred from the answer of the model. Without any fine-tuning, this can lead to a lack of prediction if the answer does not contain the label or to misinterpretation if the model formulates an answer like: "Yes, the input is not ironic". The beam search decoder is also a stochastic method and depends on several parameters.

Even if it seems more convenient to use a close answer, work of [24] claims that an open answer when correctly used, provides better results. So in this paper, we decided to test both.

3.4 LoRA

The Low-Rank Adaptation (LoRA) method, introduced by a team of Microsoft researchers in 2021 [25], has gained significant popularity as a fine-tuning technique. Widely adopted, it has proven effective for refining LLM, diffusion models, and various other types of artificial intelligence models. LoRA differs in two respects from other fine-tuning methods. First, it tracks changes to the weights instead of directly updating the weights. Second, it decomposes the large matrix of weight changes into two smaller matrices that contain the "trainable parameters." This process is called matrix decomposition. The two decomposed matrices obtained are saved and can be used by an adapter. To do this, we multiply the two matrices and add the results to the original weights of the model. This addition is often weighted by a factor generally called alpha which will control the impact of adapting it to the original model. This is how LoRA works on a layer of the model. We

[6] Free for research and commercial use but not open source

[7] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[8] https://huggingface.co/meta-llama/Llama-2-7b-hf

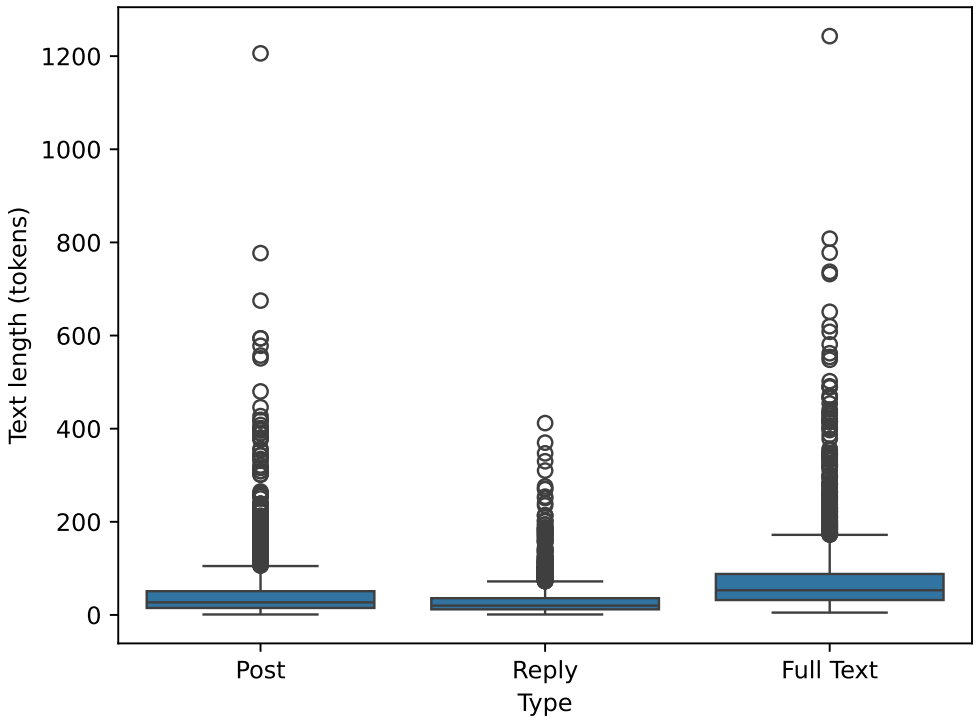

Figure 1: Distribution of text length (tokens) in EPIC

can use this technique on all the layers of our model or on only a few specific layers. Matrix decomposition can be improved by increasing the rank of our change matrices. This allows more precise tuning while keeping a fairly small number of parameters to train.

4 Dataset

4.1 EPIC

As mentioned earlier, the EPIC dataset contains 3,000 Post/Reply pairs. We label the pairs as ironic or non ironic using the majority decision and do not keep the equalities, which leaves us with 2,767 texts. The majority of the pairs in the dataset are not ironic (76%), as seen in table 1.

Label	Percentage
Non Ironic	76.545
Ironic	23.455

Table 1: Percentage of ironic and non-ironic pairs in the EPIC dataset, using the majority decision

As can be seen in table 2 and figure 1, the posts are most often longer than the replies, with some of them being very long (1,206 tokens).

	Reply	Post	Full text
Min	1	1	5
25%	12	15	32
50%	20	27	53
75%	36	51	88
Max	412	1,206	1,243

Table 2: Distribution of text length (tokens) in EPIC

4.1.1 Emojis

We hypothesize that the amount of emojis used in a post might be useful to classify it as ironic or not, as they could, for example, be used to emphasize the intention of the user. We start by looking at the average number of emojis per text (post and reply): 0.5, then the average number of emojis per non-ironic reply: 0.29 and finally the average number of emojis per ironic reply: 0.09. After testing whether the distribution of emojis in Ironic replies and Non-ironic replies differ from normal distributions (they do, respectively: p-values: 3.099e-297 and 0. We reject the null hypothesis: the samples do not come from normal distributions), we perform a Mann-Whitney U rank test on the distribution of the number of emojis in ironic and non-ironic replies and conclude that the difference in the two distributions regarding the number of emojis per reply is significant (p-value: 3.609e-05. We reject the null hypothesis: the distributions are not equal).

Examining the difference between the two sources, we find 1,457 emojis in the pairs from Twitter and 47 emojis in the pairs from Reddit. We decide to exclude the pairs from Reddit and check the average number of emojis per twitter reply: 0.51, then the average number of emojis per non-ironic twitter reply: 0.51 and finally the average number of emojis per ironic twitter reply: 0.49. We perform the normality tests (Non-ironic distribution: p-value: 2.466e-86, Ironic distribution: p-value=1.057e-287. We reject the null hypothesis: the samples do not come from normal distributions) and the Mann-Whitney U rank test: this time, we conclude that the distributions are not significantly different regarding the number of emojis (p-value: 0.36. We keep the null hypothesis: the distributions are not significantly different)

4.1.2 Hashtags

Next, we hypothesize that, like the amount of emojis, the presence and existence of hashtags could be used to classify a pair as ironic or not. As the use of hashtags in the Reddit pairs is negligible (2 out of 1,374 pairs), we exclude them again.

Unfortunately, the amount of hashtags in the Twitter corpus is also quite low (167 pairs with at least one out of 1,393 pairs). Additionally, examining the top hashtags in the corpus yields no insights as most of them seem to be used for marketing purposes (#GalaxyZFlip3, #AsharamjiBapuQuotes, #FlipAtFlipkart, #GalaxyZFlip3Giveaway, #stufflistingsarmy)

4.2 TweetEval

The TweetEval irony dataset[9] contains 4,601 tweets. As seen in table 3, the classes are more evenly distributed than in EPIC.

Label	Percentage
Non Ironic	51.923
Ironic	48.076

Table 3: Percentage of ironic and non-ironic tweets in the TweetEval irony dataset

	Tokens
Min	1
25%	14
50%	20
75%	27
Max	305

Table 4: Distribution of text length (tokens) in TweetEval

From table 4 we see that the texts in TweetEval are smaller than the ones in EPIC.

4.2.1 Hashtags

As above, we propose that the utilization and existence of hashtags could serve as indicators for categorizing a tweet as either ironic or not. 40% (1,849 out of 4,601) tweets contain hashtags. We find on average 0.664 hashtags per ironic tweet and 1.23 per non ironic tweet. We perform the normality tests (Non-ironic

[9]TweetEval Irony Dataset

distribution: p-value: 4.167e-308, Ironic distribution: p-value=0. We reject the null hypothesis: the samples do not come from normal distributions) and the Mann-Whitney U rank test and conclude that the distributions are significantly different regarding the number of hashtags (p-value: 4.161e-06. We reject the null hypothesis: the distributions are significantly different regarding the number of hashtags).

Looking at the top hashtags for ironic and non-ironic tweets show some differences: the most used in ironic tweets include *#lol* or *#fml* (which stands for *Fuck my life*) while the more frequent ones in non-ironic tweets contain *#love* or *#good*.

5 Experiments

We conducted a total of 5 experiments on the two datasets, except for the integration of LoRA which was only tested on TweetEval. The latter is an exploratory work of this project and we did not have time to adapt it for EPIC. The experiments made on RoBERTa were repeated 5 times to avoid statistical noises. The ones made on Llama2 were only ran once. For the close answer, this does not influence the results since they are deterministic. But it's clearly a weakness for the open answer and the LoRA experiments. With the poor results obtained for both, statistical noise should not influence too much our conclusion.

5.1 Data cleaning

For the EPIC dataset, we apply some preprocessing before training. we removed the pairs that were marked by an equal number of annotators as ironic and non-ironic, we removed the newlines and we replaced the emojis with their descriptions using the demoji[10] package.

For the TweetEval dataset, we remove the hashtags that indicate irony (*#irony*, *#sarcasm* and *#not*), the brackets ({ and }) and replace the emojis by their descriptions.

For both datasets, we kept only examples with less than 125 tokens. We end up with 4,599 out of 4,601 (0.99%) and 2,690 out of 3,000 (0.89%) for EPIC. Table 5 shows the different splits size for each dataset. Table 5 values may vary slightly (1 to 10 examples) depending on the experiment and repetition. For TweetEval, repetition has been made on the same splits as the ones defined in SemEval-2018. For EPIC, we made 5 different stratified splits. The stratification ensures to keep the same repetition for ironic and non ironic examples.

	Train	Validation	Test
Tweeteval	2860	955	784
Epic	1600	550	540

Table 5: Approximate training, validation, test size for both datasets.

5.2 Hyperparameters

For the MCC loss experiments, we had some difficulties finding appropriate batch size and learning rates. As the MCC is calculated on a confusion matrix, the batch size must contain enough examples to make the gradients representative. We therefore calibrated our learning rate according to the maximum batch size that we could use with our hardware. The final batch size used is 64 for a learning rate of 1e-5. The training was monitored using validation loss and an early exit configured with a patience of 5. All our experiments ended up without reaching the maximum allowed epochs which was 50.

We used the same configuration for the CE loss. We weighted the loss using the distribution of the labels in the train set.

For the open answer experiments, we used the following parameters for the decoder:

- max_new_tokens=512,
- top_p=0.9,
- top_k=50,

[10] Demoji on GitHub

- temperature=0.6,
- num_beams=1,
- repetition_penalty=1.2,

The most important is certainly the repetition penalty which for Llama2 could cause the model to repeat the prompt if it is not high enough.

For the LoRA experiments, the original paper authors claim that there is not significant difference for a rank between 8 and 258. They claim that the most important is that LoRA is applied on each layer of the original model. They also show that the alpha parameters should be set to double of the rank. But our first try using this recommendation ended up with no results.

A more recent paper [26] shows that alpha should be set between 25% - 50% of the rank. Using all the information and the documentation provided by the PEFT[11] package, we "succeed" in obtaining results using a rank of 64, an alpha of 8 and a dropout of 0.1. The parameters are still not optimal and it should be taken into account regarding the results.

6 Results

As Table 6 shows, there are no big differences between using MCC loss and CE (Cross Entropy) weighted loss. These results were expected on TweetEval given that the dataset is balanced. However, we expected the MCC loss to perform better on EPIC. We can still observe that the standard deviation on EPIC is a little higher (0.069 vs 0.021 for MCC), which tends to suggest that the MCC loss is more stable. In any case, the number of tests is too limited and the values too insignificant to draw any conclusion. Table 7 shows that unexpectedly, CE appears to converge faster than the MCC loss for EPIC. We expected the opposite results. This can surely be explained by the difficulty of using an MCC loss for this type of task as we have already discussed in section 5.2.

The behavior of the loss of functions for all models (Appendix A.2) seems to show that there are too large differences between the training and evaluation sets. We can see that in all cases, the training loss continues to decrease where the validation loss has converged.

Dataset	Model	Method	Ironic F1	Not Ironic F1	MCC
TweetEval	Roberta	MCC	0.774(0.021)	0.726(0.007)	0.520(0.016)
		CE	0.786(0.028)	0.721(0.005)	0.521(0.017)
	Llama2	Close	0.427	0.552	0.101
		Open	0.486	0.489	0.019
		LoRA	0.558	0.486	0.061
epic	Roberta	MCC	0.860(0.008)	0.520(0.023)	0.384(0.021)
		CE	0.835(0.026)	0.544(0.051)	0.388(0.069)
	Llama2	Close	0.594	0.425	0.176
		Open	0.204	0.195	-0.033

Table 6: Ironic F-score (F1), Non Ironic F-score (F1) and MCC for all experiments. For experiments with multiple runs, results are shown with this format: mean(std).

6.1 Model comparison

Overall, methods using RoBERTa perform much better than those using Llama2. This was to be expected for open and close, even if so-called zero-shot techniques can have good results on simple tasks, they rarely exceed the performance of a supervised model. The results are still very close to random which allows us quite easily to conclude that detecting irony is definitely not a simple task. We can still observe that RoBERTa obtains better results on TweetEval with the two loss functions (0.520 vs 0.384 MCC, +74%) while Llama2 obtains better results on EPIC for the close answer (0.176 vs 0.101 MCC, +57%). Even if this remains very close to random, we believe that Llama2 takes better advantage of the context provided by EPIC. Unlike

[11] https://github.com/huggingface/peft

Dataset	Method	Epoch			
		min	mean	max	std
TweetEval	MCC	13.000	19.200	24.000	4.658
	CE	16.000	20.400	24.000	3.361
epic	MCC	15.000	19.200	23.000	3.899
	CE	6.000	14.600	20.000	5.272

Table 7: Number of epochs reached before the early exit for RoBERTa experiments.

TweetEval, EPIC is annotated on two utterances. It is entirely possible that RoBERTa has more difficulty extracting features than Llama2 for this kind of context. This is not so surprising considering the differences in the number of parameters of the two models.

Regarding Llama 2 more specifically, the most promising results are actually obtained using the close answer method. For LoRA, we can observe in Figure 2 that the training and validation losses both converge around 10 epochs. After this point, losses continue to decrease very slightly with a slight rebound around 20 epochs. When we look at the results in more detail. Only 466 out of 784 (60%) of the results have a correct format. It therefore retains a large number of misinterpretations that influence the results. Examples (1) show some remaining format errors that the model produces. We don't think that increasing the number of epochs or changing the configuration will greatly improve the results. Perhaps we should use a more suitable adapter like Prefix-Tuning or Prompt-Tuning.

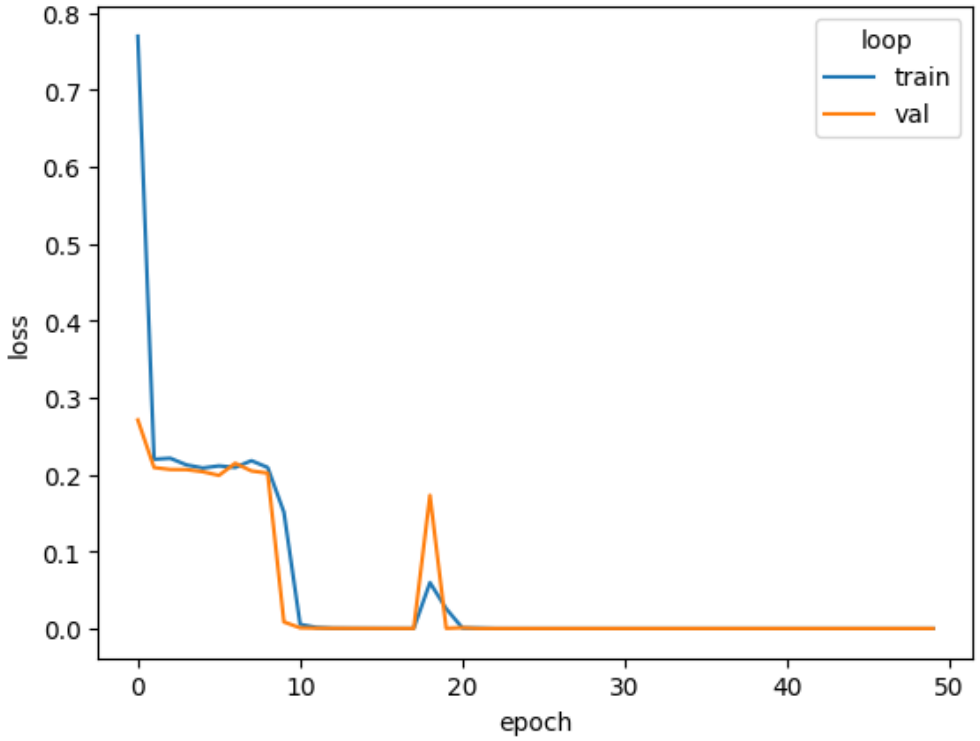

Figure 2: Training and validation loss for LoRA experiment on TweetEval

(1) a. The tweet is not ironic.
b. ' ### Output: ### Explan'
c. @user Re: Jamie Grace has T

6.2 TweetEval

For TweetEval, we investigated the results using the second annotation layer of the dataset. The table 8 shows the accuracy for each sub-category of irony for our models. We can observe that the RoBERTa models have fairly balanced results and that the differences in performance are correlated to the support of the category. For the experiments with Llama2, we can see that the model tends to predict as "ironic" all the examples. We could already conclude this by looking at the difference in F1 between "ironic" and "not ironic" in table 6. It is therefore difficult to judge if other sub-categories' accuracy is relevant. The high performances observed on Non Ironic subcategories are certainly related to this issue. Anyway, we think one of TweetEval flaws is that the subcategories are not balanced. As a result, the model learns to recognize the most represented subcategory to the detriment of the others. There are several solutions to this problem. We could try to create a more balanced dataset, try to focus on one category at a time, or even move from detection to classification.

Sub-category	MCC	BCE	Close	Open	Support
non-ironic	0.762(0.015)	0.772(0.019)	0.311	0.402	473
ironic by clash	0.778(0.013)	0.770(0.026)	0.768	0.555	164
situational irony	0.699(0.021)	0.701(0.014)	0.859	0.741	85
other irony	0.684(0.051)	0.713(0.028)	0.709	0.613	62

Table 8: Accuracy of our models for each sub-category of irony.

6.3 EPIC

For EPIC, Table 9 shows the MCC on sample results considering agreement between annotators. The agreement was calculated like this:

$$Agreement = \frac{max(|A_{i,0}|, |A_{i,1}|)}{|A_i|}$$

Where A_i is the set of annotations for a given example and $A_{i,0}$, $A_{i,1}$ are respectively the number of annotation ironic and the number of annotation not ironic for a given example.

We obviously see that the performances of the different models improve depending on the agreement. Interestingly enough, for an agreement greater than 0.9, the MCC of the model trained using the MCC loss even exceeds the MCC score obtained on TweetEval (0.526 vs 0.567). Another interesting thing is that the MCC of the model trained with BCE (Binary Cross Entropy) decreases on average for an agreement greater than 0.9 (0,506 vs 0,480). We have no explanation for this last observation. This could just be statistical noise given that we only have 5 repetitions.

Agreement	MCC	BCE	Close	Open	Support
0.5	0.384(0.023)	0.388(0.077)	0.176	-	3,000
0.6	0.390(0.022)	0.391(0.075)	0.178	-	2,750
0.7	0.479(0.034)	0.450(0.102)	0.192	-	2,038
0.8	0.526(0.026)	0.506(0.104)	0.207	-	1,673
0.9	0.567(0.055)	0.480(0.147)	0.228	-	990
1	0.567(0.055)	0.480(0.147)	0.228	-	990

Table 9: MCC for each model based on agreement score of annotators. We had a technical issue retrieving example IDs for the open experience so we can't report them in this version.

6.4 Qualitative analysis

We uploaded the classification model on HuggingFace (model), used it in a space to experiment with it and logged the results in a dataset to observe its variation in performance in different situations.

(2) a. You look nice today, like some sort of ancient creature that crawled out of the sewers.
RoBERTa: *Irony*
Llama: *Irony*

b. You look pretty today, like some sort of ancient creature that crawled out of the sewers.
RoBERTa: *Non Irony*
Llama: *Irony*

We can see that slightly changing the adjective in the sentence leads to different results, despite the overall meaning not changing.

(3) a. Cat greets mouse.
RoBERTa: *Non Irony*
Llama: *Non Irony*
b. Cat greets mice.
RoBERTa: *Irony*
Llama: *Non Irony*
c. Cats greet mice.
RoBERTa: *Irony*
Llama: *Non Irony*

Comparing examples (3-a) and (3-b), the plural influences the result, but not for all cases, in example (3-c), we can see that the plural of 'cat' did not change the result.

(4) a. You are pretty.
RoBERTa: *Non Irony*
Llama: *Non Irony*
b. You are 'pretty'.
RoBERTa: *Irony*
Llama: *Non Irony*

Punctuation can influence the results. As can be seen in examples (4-a) and (4-b), the classification changes for RoBERTa after adding the quotation marks which, in this example, makes sense for the classification to change as it is now ironic.

(5) a. CAT GREETS MICE.
RoBERTa: *Non Irony*
Llama: *Non Irony*

Comparing example (3-b) and example (5-a), capitalizing the sentence can also change the classification.

(6) a. You look nice today, like a princess. 😃
RoBERTa: *Non Irony*
Llama: *Irony*
b. You look nice today, like a princess. 💩
RoBERTa: *Irony*
Llama: *Irony*

In example (6), we add emojis to the sentences, which results in a different classification.

7 Conclusion

In this paper, we have presented the approach to detect irony. We considered two main approaches, traditional LM and LLM and trained the model based on two datasets, EPIC and TweetEval. Regarding LLMs, although our experiments failed to demonstrate this concretely, we still think that they are more suitable solutions for detecting irony. The results still allow us to shed light on certain hypotheses. Our results still allow us to clarify certain hypotheses. LL.M.s appear to benefit more from the broad background that is often necessary to interpret irony. LLMs also seem to be less sensitive to language fluctuations such as synonyms, punctuation, capitalization, etc.

Regarding the datasets, our results suggest that there is still a need to constitute corpus on irony. Our results on TweetEval show very clearly that if we want to cover the broad spectrum of irony, we need enough examples of each type. On the other hand, our results on EPIC show that irony is subjective and that it is difficult to consistently annotate this kind of corpus. We suggest two possible directions in this regard. The first consists of trying to annotate as many examples as possible in the simplest possible way (binary). The idea is that for an LLM to learn a complete understanding of irony, it needs enough examples of each. Also, having a sufficient amount of examples should mitigate the subjectivity of the annotators. The second consists of a finer approach, probably more time-consuming and therefore involving fewer examples. The idea is that it would be interesting to investigate the mechanisms of irony interpretation. To date, no corpus offers a way to understand what differentiates an ironic contradiction from a non-ironic contradiction. The approach presented in section 2.3 and the theoretical definition leave us thinking that an ironic contradiction is interpreted as the opposite of what is expressed. It would be interesting to be able to study this idea, especially in embedding spaces and attention spaces.

References

[1] Antonio Reyes, Paolo Rosso, and Davide Buscaldi. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12, 2012.

[2] María Estrella Vallecillo Rodrguez, Flor Miriam Plaza Del Arco, L. Alfonso Ureña López, and M. Teresa Martín Valdivia. SINAI at SemEval-2023 task 10: Leveraging emotions, sentiments, and irony knowledge for explainable detection of online sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 986–994, Toronto, Canada, July 2023. Association for Computational Linguistics.

[3] H. Paul Grice. Logic and conversation. *Syntax and Semantics*, 3:41–58, 1975.

[4] Cameron Shelley. The bicoherence theory of situational irony. *Cogn. Sci.*, 25:775–818, 2001.

[5] Gregory Currie. Kinds of irony: A general theory. In Raymond W. Gibbs, Jr and Herbert L. Colston, editors, *The Cambridge Handbook of Irony and Thought*, pages 17–35. Cambridge University Press, December 2023.

[6] Raymond W. Gibbs, Jr and Herbert L. Colston, editors. *The Cambridge Handbook of Irony and Thought*. Cambridge University Press, 1 edition, December 2023.

[7] Raymond W. Gibbs and Jennifer E. O'Brien. Psychological aspects of irony understanding. *Journal of Pragmatics*, 16:523–530, 1991.

[8] Julia C. Jorgensen, George A. Miller, and Dan Sperber. Test of the mention theory of irony. *Journal of Experimental Psychology: General*, 113:112–120, 1984.

[9] Dan Sperber and Deirdre Wilson. Irony and the use-mention distinction. 1981.

[10] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases? *ArXiv*, abs/1909.01066, 2019.

[11] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[12] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021.

[13] Cynthia Van Hee, Els Lefever, and Véronique Hoste. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[14] Joseph L. Fleiss. Statistical methods for rates and proportions. 1973.

[15] Simona Frenda, Alessandro Pedrani, Valerio Basile, Soda Marem Lo, Alessandra Teresa Cignarella, Raffaella Panizzon, Cristina Marco, Bianca Scarlini, Viviana Patti, Cristina Bosco, and Davide Bernardi. EPIC: Multi-perspective annotation of a corpus of irony. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13844–13857, Toronto, Canada, July 2023. Association for Computational Linguistics.

[16] Dong Nguyen and Carolyn Penstein Rosé. Language use as a reflection of socialization in online communities. 2011.

[17] Jihen Karoui, Farah Benamara, Véronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 262–272, Valencia, Spain, April 2017. Association for Computational Linguistics.

[18] Alessandra Teresa Cignarella, Manuela Sanguinetti, Cristina Bosco, and Paolo Rosso. Is this an effective way to annotate irony activators? In *Italian Conference on Computational Linguistics*, 2019.

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

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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.

[21] Daniel Loureiro, Kiamehr Rezaee, Talayeh Riahi, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and José Camacho-Collados. Tweet insights: A visualization platform to extract temporal insights from twitter. *ArXiv*, abs/2308.02142, 2023.

[22] Kumar Abhishek and G. Hamarneh. Matthews correlation coefficient loss for deep convolutional networks: Application to skin lesion segmentation. *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 225–229, 2020.

[23] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023.

[24] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

[25] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[26] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314, 2023.

A Appendix

A.1 Prompts

```
<s> [INST] <<SYS>>
You are a helpful assistant.
<</SYS>>

Classify the following tweet as ironic or not: Just walked in to #Starbucks and
asked for a "tall blonde" Hahahaha. Answer only by yes or no. [/INST] The answer is

```

Figure 3: Prompt example for TweetEval close.

```
<s> [INST] <<SYS>>␍␊
You are a helpful assistant.␍␊
<</SYS>>␍␊
␍␊
Classify the following tweet as ironic or not: Just walked in to #Starbucks and
asked for a "tall blonde" Hahahaha. Answer only by yes or no. [/INST]␍␊
␍␊
```

Figure 4: Prompt example for TweetEval open.

```
<s> [INST] <<SYS>>␍␊
You are a helpful assistant.␍␊
<</SYS>>␍␊
␍␊
Below is a dialogue between user A and user B.␍␊
␍␊
User A:@FymmG Hi I wanted to ask if we can still email for the DLC?␍␊
User B:@sick20034 Yes you can. ;)␍␊
␍␊
Is B's answer to A ironic? Answer by yes or no. [/INST] The answer is␍␊
```

Figure 5: Prompt example for EPIC close.

```
<s> [INST] <<SYS>>␍␊
You are a helpful assistant.␍␊
<</SYS>>␍␊
␍␊
Below is a dialogue between user A and user B.␍␊
␍␊
User A:@FymmG Hi I wanted to ask if we can still email for the DLC?␍␊
User B:@sick20034 Yes you can. ;)␍␊
␍␊
Is B's answer to A ironic? Answer by yes or no. [/INST]␍␊
```

Figure 6: Prompt example for EPIC open.

```
<s> [INST] <<SYS>>␍␊
You are a helpful assistant.␍␊
<</SYS>>␍␊
␍␊
Below is a dialogue between user A and user B.␍␊
␍␊
User A:@FymmG Hi I wanted to ask if we can still email for the DLC?␍␊
User B:@sick20034 Yes you can. ;)␍␊
␍␊
Is B's answer to A ironic? Answer by yes or no. [/INST]␍␊
```

Figure 7: Prompt example for TweetEval LoRA.

A.2 Loss

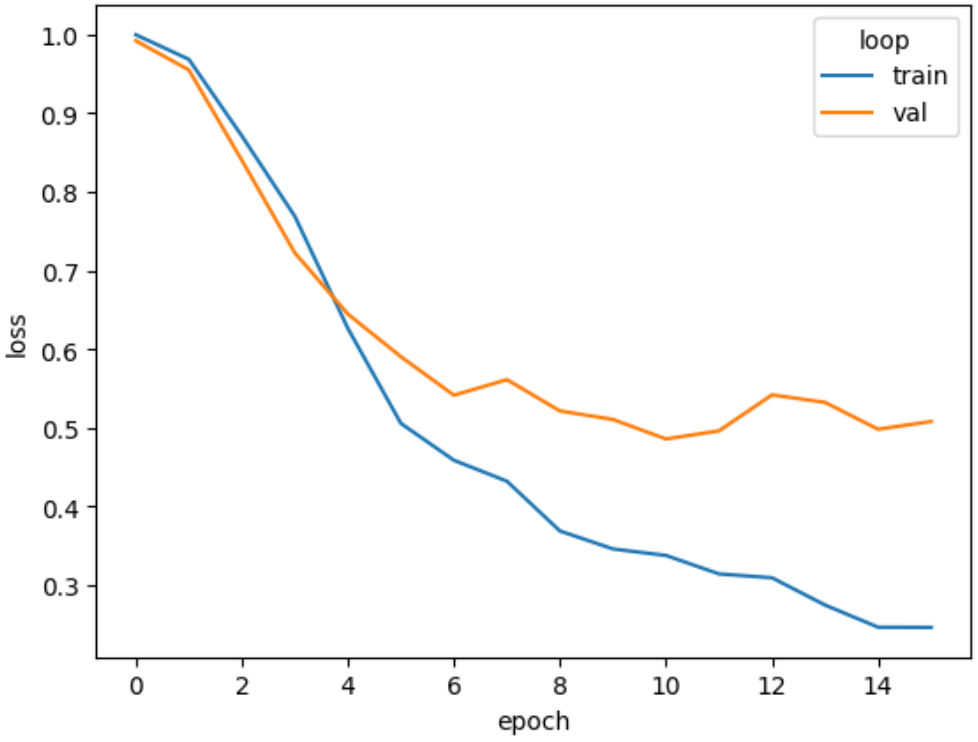

Figure 8: Training and validation loss for MCC Loss on TweetEval

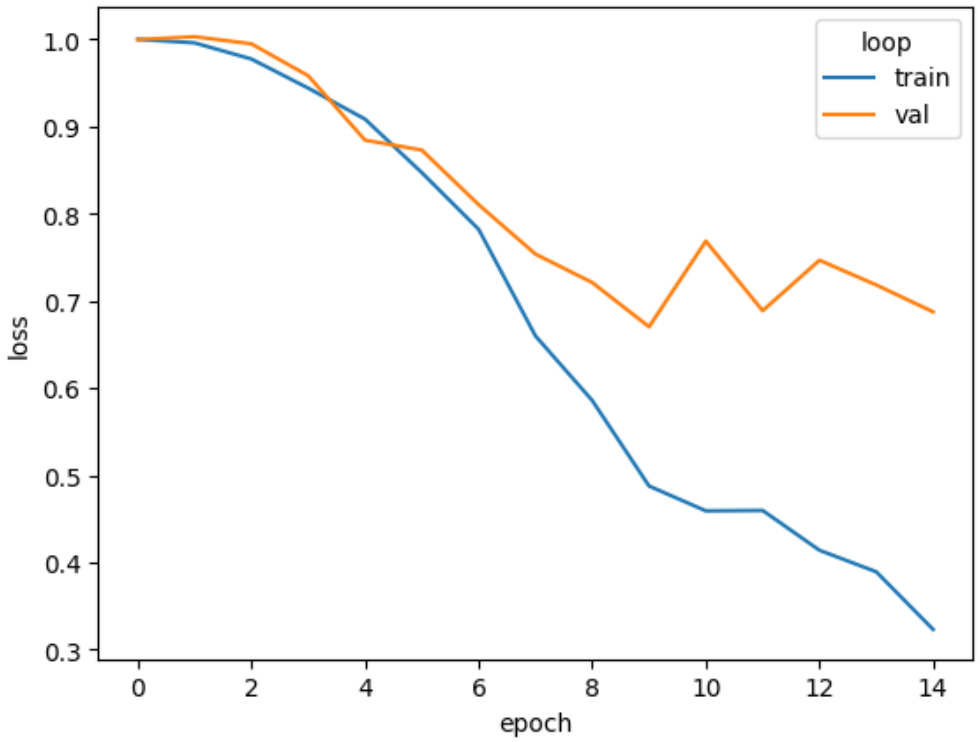

Figure 9: Training and validation loss for MCC Loss on EPIC

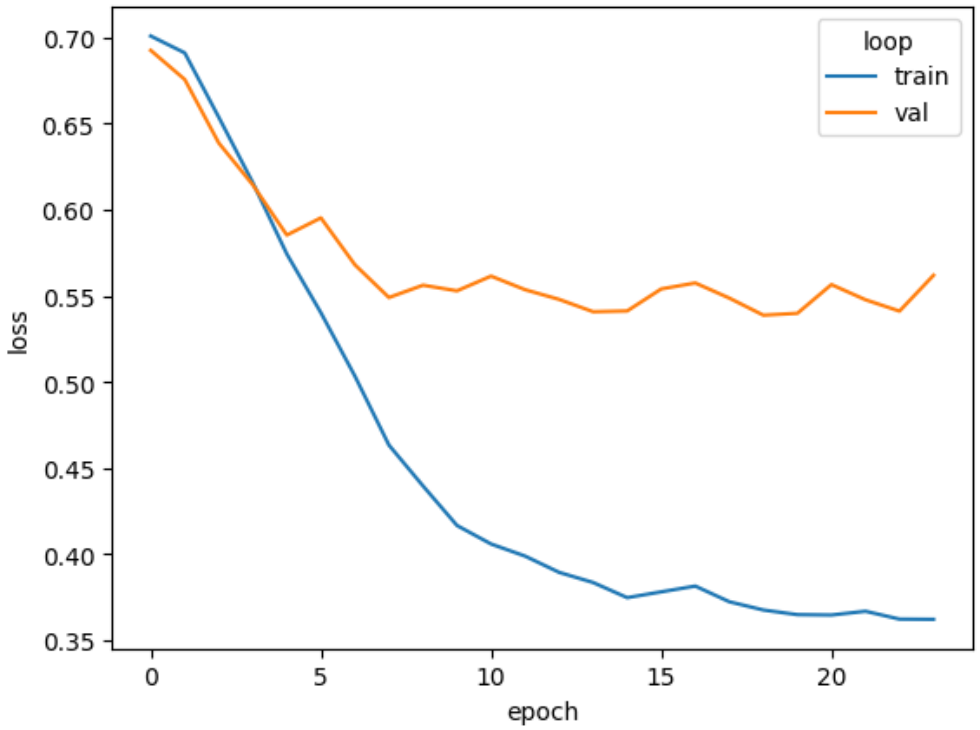

Figure 10: Training and validation loss for CE Loss on TweetEval

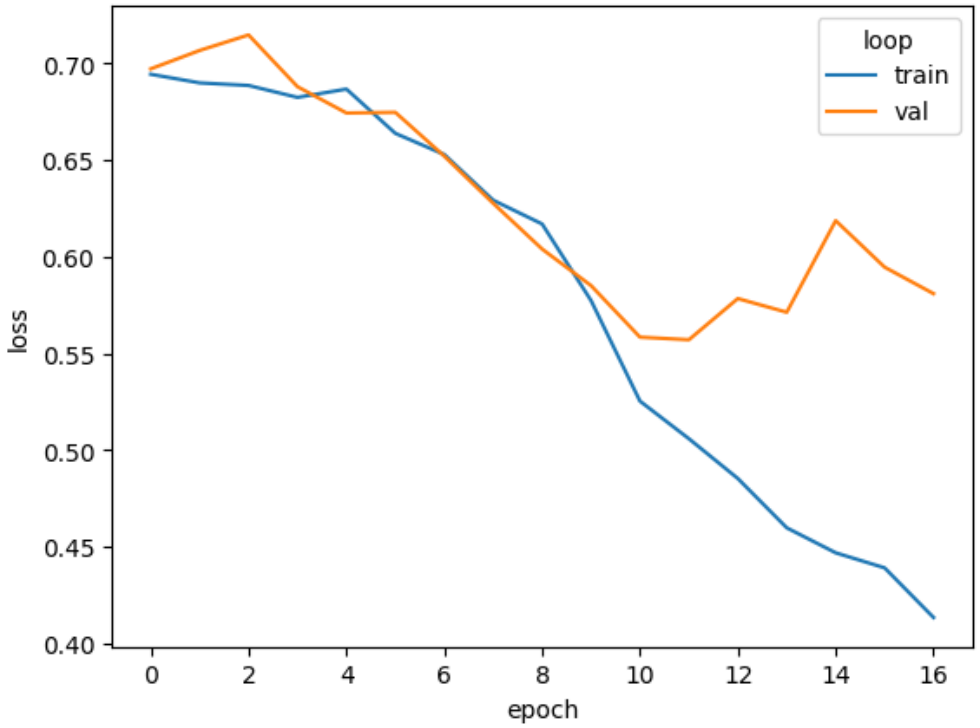

Figure 11: Training and validation loss for CE Loss on EPIC