SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations

Josef Collins[1], Miles Mena[1], Aditya Thaker[1]

[1]University of Colorado Boulder

Abstract

In text, especially dialogues, phrases from sentences combine to create an emotion in sentences that follow. The task of extracting phrases from text that create an emotion in subsequent phrases is called emotion cause pair extraction analysis. In this paper we apply a conditional random field, fine-tune two BERT models, and to the task outlined in SemEval-2024 Task 3.

1 Introduction

SemEval is an international natural language processing workshop that produces yearly sets of semantic evaluation tasks for computational systems. Additionally, they produce high quality datasets for natural language processing research. These shared tasks are based on competition and projects that contribute to the understanding of natural language processing are accepted to conferences and published. Here we propose solutions to Subtask 1 of Task 3 from SemEval-2024.

Task 3 focuses on the extraction of emotion causes from text and video gathered from the television series Friends. Subtask 1 makes use of only pre-labeled textual data, and doesn't use any video data. The task can be described as follows: given a scene containing a dialogue between a couple of characters, determine the sequence of phrases that is causing the emotion of a target sentence.

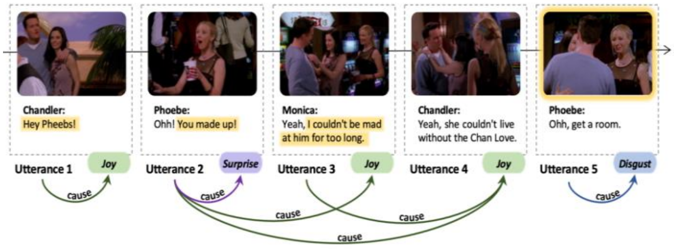

Figure 1: An example of the task and the dataset. Each arc points from the cause utterance to the emotion it triggers. The cause spans have been highlighted in yellow. Background: Chandler and his girlfriend Monicaq walked into the casino (they had a quarrel earlier but made up soon), and then started a conversation with Phoebe.

Understanding the cause of an emotion has major implications in the development of intelligent computational systems. Business analysis, computer human interaction, sentiment analysis, and chatbots are all examples of systems that improve with a knowledge of what causes an emotion.

2 Related work

Recognizing the importance of emotion cause analysis, (Wang et al., 2021), developed a dataset of labeled conversations with corresponding emotional causes. Additionally, the authors provide a baseline for the multimodal task of extracting emotion causes in their provided dataset. They adapt the model ECPE-2 steps from (Xia and Ding, 2019), and with the three data formats (textual, audio, video) they achieve an overall f1 score of .5132.

The task of emotion cause analysis in conversations has been shown to improve when coupled with the task of emotion recognition (Chen et al., 2022). A multi-task recurrent synchronization (RSN) network of long-short term memory (LSTM) networks allows for information across tasks. As the RSN recognizes emotions and extracts phrases that cause the emotion it learns information that is unique to each task. Then the RSN combines these two tasks together to perform them at the same time, with the specialized information from each of the tasks.

BERT is a state of the art model that consistently performs well on several tasks. (Devlin et al., 2018). BERT stands for Bidirectional Encoder Representation from Transformers, and it encodes texts by conditioning on both directions. A masked language model is applied in the pre-training process so that BERT can learn encodings without having knowledge of the exact word it's learning.

3 Models

We apply three different models, conditional random fields, BERT, and transformers. We apply BERT with two different configurations. The first trains Bert at the utterance level and uses two classifiers at the same level to label the sequence. The second uses BERT to generate token embeddings for dialog samples.

3.1 Conditional Random Fields

Conditional Random Fields (CRFs) are considered effective for sequence modeling tasks in Natural Language Processing (NLP) due to several key advantages. CRFs naturally capture contextual dependencies, making them suitable for incorporating sequential dependencies. CRFs consider the entire sequence when assigning labels to individual elements. This global perspective helps maintain consistency across the entire sequence and ensures that the labels assigned to neighboring elements are coherent. CRFs allow the incorporation of rich feature representations. Features can capture a wide range of information, including word identities, part-of-speech tags, and other relevant linguistic features.

The utterances in a dialogue lead up to the target utterance for which we want to find the cause span. This can be viewed as a sequence labeling task which is used to indicate if the current utterance is a part of the cause span or not. We model this task as a sequence labeling from the label set {0,1}.

For this model, we use different types of vector embeddings to compare the model results. The first method is to simply concatenate the utterance, author and emotion into one document. We then use Doc2Vec to get a vector representation of this sentence. The second method is to use the Doc2Vec representation of the utterance, then perform Word2Vec on the emotions and author names. These 3 vectors can then be aggregated into a final embedding vector. We use different aggregation methods like, sum, average and max pooling to get a total of 4 different embedding vectors.

The model is run on these 4 embedding vectors and a comparison of different training regimes is shown. The first method is "lbfgs" which is basically a modified gradient descent algorithm. The L-BFGS method is a type of second-order optimization algorithm and belongs to a class of Quasi-Newton methods. It approximates the second derivative for the problems where it cannot be directly calculated.

The second training algorithm is Stochastic Gradient Descent with L2 regularization. The third and last one is average perceptron, which is an early and simple version of a neural network. In this approach, inputs are classified into several possible outputs based on a linear function, and then combined with a set of weights that are derived from the feature vector—hence the name "perceptron."

The results of this model are shown in Table 4.1 and 4.2.

3.2 BERT on each Utterance

The data of this model is split .8 for training, .1 for validation, and .1 for testing. Additionally, the training data is formatted so that each sentence for every conversation is paired with the targeted phrase of that conversation. So utterance ID 1 is paired with the utterance we are extracting the emotion causes from.

These pairs are eventually concatenated and encoded with BERT. Here is a sample of the decoded pairs: [CLS] No . [SEP] Hi ! I am Dr. Drake Remoray and I have a few routine questions I need to ask you . [PAD] [PAD]... . In this case the label is [0,0], since no part of the phrase [No .] is in the cause span.

BERT is a pre-trained model that has learned encodings of words from previous data. For this training we freeze those parameters so that the only thing that is being learned are the weights of the classifier.

To turn the model into a sequence labeler, we pass the hidden layer outputs into two classifiers at the same level, where one classifier learns to predict if a word is in the cause span and the other predicts if the word is out of the cause span.

We then stack the output of the classifiers and apply the log softmax. For learning the weights of the classifiers we back propagate the Natural Log Loss with AdamW.

Results are listed in the table below. We used 100 epochs, an output size of 64, learning rate of .001, and a hidden size of 32.

3.3 BERT Embeddings with Transformer

Data was prepared by concatenating utterances on a per dialog basis. Each utterance was separated by a separator special token. Custom special tokens for each of the seven possible emotions, including neutral, were prepended before the start of each utterance. The corresponding utterance for each dialog's emotion utterance id was enclosed by custom special tokens indicating the target start and target end. All of these special tokens were added to the BERT tokenizer. With this dataset, token embeddings were generated for each sample. This produces a large tensor, since each token embedding is BERT's standard 768 dimensional vector.

Labels for each sample were generated on a per token basis. If the token at a given index in the sample fell in one of the emotion utterance id's cause spans, that label was assigned either B or I classes at the same index. B indicating that the index was the start of the cause span and I indicating that it was included in a cause span. Indices outside of a cause span were assigned the O class.

Both samples and labels were padded with the built-in BERT pad token. Attention masks were generated for later use in the transformer model. True in the mask indicates that the content at that index is padding and can be ignored during gradient calculations.

Once encoded, samples and labels were divided into training and dev sets. Batches of size 32 were passed into a transformer encoder model. Cross entropy loss was utilized as well as Adam optimization.

4 Results

The table below shows the different F1 scores for the embeddings trained using different training regimes.

Algorithm	Doc2Vec	Word2Vec Add	Word2Vec Average	Word2Vec Max
AP	0.853	0.853	0.853	0.853
LBFGS	0.856	0.856	0.856	0.856
L2SGD	**0.857**	**0.857**	**0.857**	**0.857**

The table below shows the fraction of times the whole sequence matched with the target sequence.

Algorithm	Doc2Vec	Word2Vec Add	Word2Vec Average	Word2Vec Max
AP	**0.339**	**0.339**	**0.339**	**0.339**
LBFGS	0.320	0.320	0.320	0.320
L2SGD	0.320	0.320	0.320	0.320

The following table describes the F1 scores on unseen testing data for each of the three approaches.

Model	Testing F1
CRF	.85
Bert Utterance Level	.44
Bert Embeddings with Transformer	0.77

5 Discussion

Some of the challenges that arose during development were related to the dimensionality of BERT. Concatenating utterances provides more in depth context when generating embeddings. However, this method runs the risk of too long sequences. The maximum sequence length BERT is able to handle is 512. This requires that sequences over 512 tokens are truncated, causing data loss. Fortunately the vast majority of dialogs were fewer than 512 tokens after tokenization, but a different method would be needed for embedding generation on larger dialogs.

Embedding tensors generated by BERT took up extensive memory. This was compounded by the fact that dialogs were padded. One solution to this problem would be to preprocess all dialogs and remove any words that were unlikely to provide useful context for identifying the cause spans. These tokens would, in most cases, come after the emotion utterance's id's respective utterance. In some rare cases, dialog after the target utterance was included in a cause span. However, in most cases this following dialog could have been excluded. This would drastically improve memory usage and likely improve training speed and accuracy.

Additionally, the performance of BERT at the utterance level justifies the training method of BERT Embeddings with Transformers. When training at the utterance level, the model learned to greedily label most elements at the beginning as belonging in the cause span. For sequences that don't belong in the cause span the model performs extremely poorly, and predicts similar to a sequence that has parts within the cause span.

The embeddings used for the CRF model show very little variance in the results. This could be attributed to the fact that the majority of the weight for the vector models comes from the encoded utterances using Doc2Vec. The embeddings that only use Doc2Vec on concatenated data were initially considered as something that would pay lesser attention to the words at the end which were important to identify speaker and emotion. However, it is possible that due these words being repetitive and very closely related to each other, aggregating them with a Doc2Vec embedding of just the utterance, doesn't make a big difference in the embedding.

The training algorithms were mainly divided into 2 parts: the averaged perceptron and the gradient descent learning. The gradient descent algorithms provided a better F1 score but fell short on the whole sequence matching task, the average perceptron did just the opposite. The overall results of the model are very robust and do not change very much based on the training regime selected and the hyperparameters.

Acknowledgments

Thanks to Dr. James Martin, Jie Cao, and all the wonderful TA's of CSCI-LING 5832

6 References

Chen, F., Shi, Z., Yang, Z. and Huang, Y., 2022. Recurrent synchronization network for emotion-cause pair extraction. *Knowledge-Based Systems*, *238*, p.107965.

Wang, F., Ding, Z., Xia, R., Li, Z. and Yu, J., 2021. Multimodal emotion-cause pair extraction in conversations. *arXiv preprint arXiv:2110.08020*.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lafferty, J., McCallum, A. and Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

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