Project Midterm Report

Joshua Mares[1], [1]**University of California, Los Angeles, joshuamares180@gmail.com**

1 Introduction

The goal of this project is to leverage pretrained models (e.g. BERT, RoBERTa) and tune them such that they can determine if a statement is commonsensical. This project can be split into three main steps. 1) Data Processing and Loading of the SemEval and Com2Sense data sets. 2) Model Building leveraging existing libraries. My model of choice being DeBERTa. 3) Model Training and Evaluation via finetuning, transfer learning, and pair-wise accuracy metrics.

2 Methods

2.1 Datasets

The two data sets I used were the Com2Sense data set and the SemEval-2020 Task 4 data set. The Com2Sense data set was specifically created to train and test a models ability to determine if a sentence is commonsensical i.e. if it makes sense according to a humans understanding of the language and the world. It accomplishes this by having complementary sentences that differ by only a few words which invert the meaning of the sentence. The SemEval data set has a similar construction with complementary sentences as well as the reasons as to why a sentence does not make sense. For this project, however, we only use the complementary sentences to train our models.

2.2 BERT, RoBERTa, and DeBERTa

The main model I used was the DeBERTa language model which is based on the BERT and RoBERTa language models. BERT, or Bidirectional Encoder Representation from Transformers, uses a series of Encoders that use positional encoding and self attention matrices that allow the model to encode positional and lexical context to each word in a sentence. Thus while it takes in a generalized word embedding, it is able to give context and meaning to the word by its position and surrounding words. Thus, we train the model to first understand words and their contexts and then fine tune the models to our specific task, text classification. RoBERTa, or Robustly Optimized BERT approach, is a variant of BERT which aims to optimize the model. Mainly, when pre-training BERT we would mask certain words in each sentence and have BERT train over the same masked sentences for many epochs. This meant BERT would only learn how to predict the those specific masked words. RoBERTa however uses dynamic masking to change the masked words for each epoch and thus force itself to predict new words for every epoch. In addition, RoBERTa also dropped the next sentence prediction portion of its pre-training step. Furthermore, RoBERTa increased it text corpus, batches, and sentence sequence lengths to improve performance. Finally, DeBERTa, or Decoding Enhanced BERT with Disentangled Attention, is an extension to BERT and RoBERTA that seperates the positional encoding and the word encoding. In BERT and RoBERTa, the model would take in the raw word embedding and the raw positional embedding and combine them into a single vector that is then contextualized by the model. In DeBERTa however, we instead keep the two embeddings seperate and use them individually within the model. Where each position-encoded-word had a single query and key vector in BERT and RoBERTa, each word and position vector have their own Query and Position vectors in DeBERTa. This allows us to "disentangle" the attention matrices created by multiplying the key and query vectors. Now, we can create an attention matrix between each key and query. So we will have a word-to-word attention matrix, a position-to-position matrix, a position-to-word attention matrix, and finally a word-to-position attention matrix. This allows us to determine how much the each word influences each other, how much each posi-

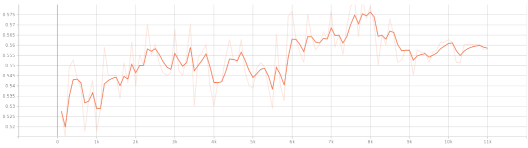

Figure 1: Transfer Learning: Accuracy

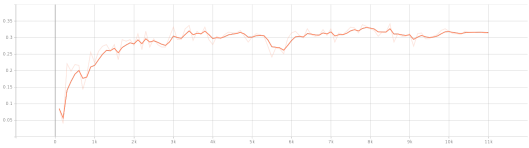

Figure 2: Transfer Learning: Pairwise Accuracy

tion influences each other, and how much the each position influence the surrounding words and vice versa. However, since the positional encodings are relative the position-to-position matrix is not used. However, the lack of absolute encodings may lead to ambiguity when looking at words with similar nearby surrounding words. Thus, the model still uses absolute position embeddings, but at the very end as to minimize their influence on the attention matrices and instead allow the model to focus on the relational influences. DeBERTa is currently the best performing model of the three, so chose to use it over BERT and RoBERTa.

2.3 Open Ended Task

For the open ended portion of this assignment, I chose to do transfer learning. Transfer learning is the process in which we train a model on a massive corpus for one task. We then use the weights of that model as a starting point for another model that will be trained on a different task with a smaller corpus. This allows the second model to use any information learned by the first and apply it to the new task. In this case, I will first train a DeBERTa model on the SemEval data set since it is a similar problem but with a much larger corpus. I will then initialize a new DeBERTa model with the weights from the SemEval trained model and then train the model on the Com2Sense

2.4 Metrics

We will calculate F1-Score, accuracy, loss, precision, and recall when training and evaluating. If training for the Com2Sense task, we also keep track of the pairwise accuracy. Pairwise accuracy is percentage of complementary sentences the model is able to correctly for both sentences. This metric allows us to determine if the model is able to correctly apply contexts to words and thus determine related topics such as the ”fear of height” and ”ferris wheels.” Thus we will primarily use accuracy and pair-wise accuracy to gauge the models performance.

3 Results

The DeBERTa model trained exclusively on Com2Sense achieved a accuracy of 0.5565 and a pair-wise accuracy of .2714 on the development data and a accuracy of .5435 and pair-wise accuracy of 0.2642 on the test set. The model initialized via transfer learning achieved a max accuracy of 0.5829 and a max pairwise accuracy of 0.3417 on the development data and a accuracy of 0.5353 and a pair-wise accuracy of 0.2817 on the test data. Its learning process is shown in figure 1 and 2.

4 Discussion

One important thing I would like to note is that the base DeBERTa model metrics without transfer learning were provided to me by my TA who ran my code on his google cloud instance as my google cloud was not working at the time. However, I was unable to reproduce the results on my instance despite running the exact same script and arguments. I was however, able to run the transfer learning models on my own instance. Despite the transfer learning, I was not able to make a great improvement on the base metrics provided to me by my TA and was just barely able to pass the baseline requirements. There are two reasons I believe this may be. The first is that I did not train the model with enough data for it to properly learn the proper contexts for each word. This is due to my own underestimation of how much training the model would actually need as well as my own issues with training in the google cloud environment. Second, given the plateau in the pairwise accuracy, I was not able to fine tune the model correctly and thus fell into a local optimum that the model was not able to escape. If attempted again, I would attempt to properly fine tune the model as well as provide a far larger number of epochs so that the model may properly learn the contextualizes meanings of each word in the data.

5 References