Adversarial Natural Language Inference

Homework 2

Abstract

In this task, I aim to train a transformer-based model to address a standard Natural Language Inference (NLI) task using the provided dataset. Additionally, I evaluate the model against an adversarial test set. To enhance the model's performance, I implement several architectural modifications and analyze the outcomes. Furthermore, I create an adversarial training set by augmenting a subset of the original dataset, training another model on this augmented set, and subsequently assessing its performance.

1 Dataset Description

The dataset that was provided to us contains a premise, hypothesis, label and wsd, srl annotations, as you can see in the figure 1 in appendice section. The dataset features three types of labels—entailment, neutral, and contradiction—making it a multi-class classification problem. Our model needs to predict of the hypothesis is contradicting the premise or agreeing with the statement.

2 Description

First, I selected two transformer-based model to start my training on provided dataset namely "Roberta" and "distilbert_uncased". Both model was giving almost same result on standard test set and adversarial set but "roberta " model took way more time during training, each epoch was taking almost 32 minutes while "distilbert" model was taking about 24 minutes. So, we selected "distilbert" model afterward for further experimentation. For training a model, first we select the batch size of 16 because that's the best my machine could handle. We are also supposed to feed the input data which is mainly hypothesis and premise into the format as "[CLS] P [SEP] H". but since Bert models already handle these token internally, we don't have to do it. During preprocessing the data for tokenization, we also set truncation = true and max length = 128. For evaluation, we used several matrices, like accuracy, precision, f1_score, recall and auc_roc.

3 Architectural Changes

To introduce meaningful architectural changes into our model for the NLI task, I consider several approaches, such as integrating a pretrained sentence embedding to enhance our input representation which was imported from the sentence_tranformer library. Also, I add an attention Mechanisms into our model by adding an attention layer to focus on important parts of the input. So basically the model processes the input tokens (input_ids) and their attention masks (attention_mask).It extracts the [CLS] token representation from DistilBERT, which is typically used for classification tasks. Sentence embeddings are generated using the Sentence-Transformer model. The projected output from are passed through a multi-head attention layer. The attention mechanism helps the model focus on different parts of the sentence embeddings, potentially capturing more useful information. The outputs from the attention layer and the projected DistilBERT output are concatenated. The combined output (768 dimensions) is passed through the final classifier layer to produce logits for classification.

4 Generating Adversarial Dataset

In this task also have to deal with adversarial-NLI, so modifying the data in a way that the model is more robust for NLI task, so we are asked to generate adversarial data using the original dataset. In order to do that, I used several techniques all combined together to generate my own adversarial dataset comprising of 15000 samples. The focus was on augmenting the hypothesis of each sample, as this is less complex than modifying the premise. The data augmentation techniques uti-

lized include **antonym** substitution, **synonym** substitution, and **hypernym** substitution, facilitated by the Word Sense Disambiguation (WSD) annotations provided in the original dataset. For antonym substitution, the label was also switched (e.g., contradiction to entailment and vice versa). Additionally, I used **cross-lingual augmentation** technique, involving back-translation, which proved to be the most sophisticated method in preserving the original sentence's semantics, but at the cost of increased processing time. The other techniques that I used were very generic because using semantics alone may or may not be beneficial, so I used some non-semantic augmentation but in a very small percentage. One such technique was **word-swap**, where subjects or objects within a sentence were exchanged, resulting in slight disorganization without losing semantic meaning. Finally, I used **Noise insertion** technique which simply finds random words or letters in each sentence and adds them again in a sentence. This technique can help the model to be more robust to any typos that may be present in the data. We also noticed that there were many cases where some technique might not work for a sentence, in that case, I simply used again the synonym substitution technique which was way more reliable if any technique failed. Each technique was assigned a specific percentage to determine its application across the subset of 15,000 samples, as detailed in the accompanying Table 1 in the appendices. The techniques assigned a lower percentage were chosen as such due to their higher likelihood of causing a loss of semantic meaning in the sentence. The augmenting took around 3 hours to process for 15000 samples, which were then combined and shuffled across the original training dataset

5 Results

To comment on the results, I recorded accuracy, precision, f1_score, recall, and auc_roc for all experiments.

5.1 Standard Model

As mentioned before I also experimented with Roberta and distilbert model but bothe were giving same accuracy but Roberta was taking way more time during training so I selected distilbert for further experiments. The result that we got by training a standard distilbert_base_uncased model we got an accuracy of 70.2% on standard testset and 51.6% accuracy on the adversarial testset which was expected since adversarial was designed to fool the model but results were satisfactory.

5.2 Model with Architectural Changes

The result we observed after training the model that was introduced by several meaningful architectural changes was fine but not expected, the accuracy was the same 70.3% for the standard testset and 49.8% for the adversarial test set. The results were almost the same as the standard model which was not expected since this model took twice the amount of training time when compared to the standard model.

5.3 Model Trained with Adversarial Training Dataset

The result for this model were quite disappointing, since we increased the training set with an additional 15000 samples it should have performed better than the previous model but in fact the figure remained almost the same for both testset and adversarial test set, it increased in the points but we are not sure if that's random or might increase with possibly more adversarial data.It was also observed that when training the model for more than one epoch it was overfitting on the adversarial test set hence reducing the accuracy. You can see all the metrics figures in the table at the end.

6 Appendices

A Table and Figures

```
{
    "id": The ID of the sample,
    "premise": The context to be used in making the inference,
    "hypothesis": The claim that is made concerning the premise,
    "label": The annotated label for the claim.
             One of ENTAILMENT|CONTRADICTION|NEUTRAL,
}
```

Figure 1: Dataset Description

Table 1: The techniques assigned a lower percentage were chosen as such due to their higher likelihood of causing a loss of semantic meaning in the sentences

Augmentation Techniques	Percentage
Synonym, Hypernym Substitution	50%
Antonym Substitution	10%
Words Swapping	10%
Noise Insertion	10%
Back-Translation	20%

Metrics for standard test set					
Model	**Accuracy**	**Precision**	**Recall**	**F1_score**	**AUC_ROC**
Roberta_base	69.4%	68.7%	69.4%	68.1%	84.3%
Distilbert_uncased	70.2%	70.1%	70.2%	69.2%	85.4%
Distilbert (Architectural Changes)	70.3%	70.1%	70.3%	69.5%	85.5%
Distilbert (Trained on adversarial Dataset)	69.6%	70.2%	69.6%	68.4%	85.6%

Metrics for Adversarial test set					
Model	**Accuracy**	**Precision**	**Recall**	**F1_score**	**AUC_ROC**
Roberta_base	51.0%	51.6%	51.0%	51.0%	67.1%
Distilbert_uncase	51.6%	54.1%	51.6%	51.8%	65.7%
Distilbert (Architectural Changes)	49.8%	51.1%	49.8%	50.0%	65.2%
Distilbert (Trained on adversarial Dataset)	50.4%	53.3%	50.44%	50.5%	65.7%

Figure 2: Results of all the models trained