|  |
| --- |
| **Improving Neutrality-Detection in Electra-Small Models Through Dataset Cartography and Contrastive Training** |
|  |
| Siri Bafna & Nithin Anumala  SB66495 & NSA552  [siri.bafna@utexas.edu](mailto:siri.bafna@utexas.edu), nithin.anumala@utexas.edu |
|  |
|  |
|  |
|  |

Abstract

Our project consists of detecting the shortcomings of the Electra-small baseline model that has been provided by the starter-code, which then allows us to implement several modifications and fixes that can enhance the performance of the model. For this project, we used Dataset cartography, which aided in detecting hard-learner examples, as well as contrastive training. We dive into the many experiments and improvements we made to help better detect neutrality within examples. Through the medium of this project, we demonstrate how these given suggestions and fixes aided in the improvement of the model, and could be used as a method of better understanding and aiding different biases, hallucinations and minor sentence structure fixes helps in creating a better state-of-the-art model altogether.

Introduction

In today’s day, as NLP models get more and more intuitive, their necessity for having a strong, detail-oriented approach gets all-the-more essential. This is why, models need to have the ability to figure out and examine when statements are slightly off, modified, or negated, and be able to successfully capture that through the medium of a several different approaches. Through the medium of this project, we explored the several sensitivities, biases and other areas of improvement of the Electra-small model that was provided in the starter code. In this section, we will plan on covering the many areas of improvement that we noticed in the model prior to making or proposing any fixes.

**1.1 Importance of Contrast Sets**

When examining the many approaches we could’ve used for this dataset, ranging from checklists, to statistical tests, research showed that contrast sets are a pretty significant and robust methodology to recognize the model’s ability to detect sensitive, linguistic changes. Additionally, it also tests the ability for a model to generalize and properly fit to the sentences beyond what was simply trained on. Therefore, our approach for doing analysis of this data, was to use contrast sets on evaluation.

**1.2 Model’s Ability to Understand Subtle Changes**

We were able to take this model and train on the Stanford-NLI dataset. We then performed a two-way evaluation of the performance of the baseline model by evaluating it on another subset of the Stanford-NLI dataset, and a manually-created contrast set, that took some of the rather hard-to-learn aspects of the initial dataset, and made subtle changes which allowed us to examine the model’s ability to handle sensitive changes, or even negations appropriately.

**1.3 Performance Success Patterns**

While looking in-depth at the models’ evaluation results, we noticed the several patterns of success and failures it has in its contrast sets. The electra-small model does significantly well with contextual accuracy, contradictions, and neutral labeling. These can be represented with examples where there is a clear connection (positive or negative) from the premise, or in other words, the context has been accurately given and detected. The example: *Premise:* "Two women are embracing while holding to go packages." *Hypothesis:* "The sisters are hugging goodbye while holding to go packages after just eating lunch." *Label:* 0 (entailment), *Predicted Label:* 0  shows that despite not having that many words in common, it is able to detect the commonality and context between “women” and “sisters,” and “embracing” along with “hugging”. In this example, the correct label was neutral, and the model also correctly predicted this. Additionally, the model also does a great job in understanding contradictions between the premise and hypothesis and has a rate of properly detecting these contradictions at least 70% of the time. Lastly, the model does a great job in detecting neutral statements between premise and hypothesis, as shown by the following example: Premise: Two women are embracing while holding to-go packages.” Hypothesis: “Two women are holding items and shaking hands.” In this example, the correct label was neutral, and the model also correctly predicted this. Through this, we can see its ability to be able to detect neutrality, pretty well as long as it has some common words to go off of it.

**1.4 Performance Failure Patterns**

The model has some significant areas of improvement as well. While it does a good job detecting neutrals correctly, when the wording or context around a hypothesis changes, it is unable to detect the neutrality, and often ends up predicting a neutral example as entailment. An example of this is shown by the following: *Premise:* "Two women are embracing while holding to go packages." *Hypothesis:* "Two women are holding items and shaking hands." *Label:* 1 (neutral) *Predicted Label:* 0 (entailment). This is a case where the premise and hypothesis are related but not exactly equivalent, and the model incorrectly classifies it as entailment. In addition to that, at times it overgeneralizes a contradiction as well. An example of this is: *Premise:* "A man in a blue shirt standing in front of a garage-like structure painted with geometric designs." *Hypothesis:* "The man is holding a tool and painting." *Label:* 0 (entailment) *Predicted Label:* 2 (contradiction)

The model misinterprets the scenario as a contradiction, even though the hypothesis doesn't directly contradict the premise—it just adds new information that isn't strictly supported by the premise. With these examples, we can see that in obvious cases, or “easy-to-learn” methods, the model does a good job in identifying those. However, as soon as it gets to performing on neutral, or more subtle examples that have slightly changed hypotheses, it is unable to detect neutrals and rather goes to entailments or contradictions incorrectly.

Proposal

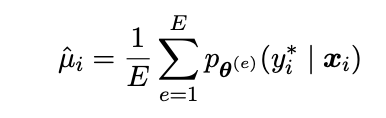
After doing rigorous analysis and categorizing the model’s issues, we now see that it does a good job in figuring out statements based on a given context, despite being a pretty base-level model. However, when things get complicated, and the model is challenged with contrast sets, it is unable to detect the labels that surround neutrality, and rather goes towards entailments and contradictions without having a strong basis for why. Therefore, with these improvements in mind, we propose that through Dataset cartography, which allows us to learn on hard subsets of the data, where neutrality might be a bit harder to label, along with contrastive training, which would allow the model to perfect subtle changes in hypotheses, will both allow for a comprehensive approach to fixing the model’s issues on neutrality-detection and contrast-sets.

Dataset Cartography

Pre-trained models such as ELECTRA have revolutionized natural language processing (NLP) by achieving high performance on benchmark tasks. However, their success often depends on the composition of the training datasets used. Certain data points may disproportionately influence the learning process, either because they are inherently easier to learn or because they introduce unique challenges due to their complexity or ambiguity. This variability in dataset composition can significantly affect a model’s ability to generalize, especially in scenarios where the test data distribution differs from the training data.

Our implementation explores the relationship between training data composition and model performance, focusing on the hypothesis that hard-to-learn examples, while challenging, contribute more to model generalization. These examples force the model to develop a deeper understanding of the task and avoid overfitting to superficial patterns. To operationalize this hypothesis, we employ Dataset Cartography (Swayamdipta et al., 2020), a framework that uses confidence scores as a measure of the ease or difficulty of learning specific data points. By classifying training examples into easy and hard categories based on these scores, we aim to optimize the proportions of these subsets in training datasets to enhance model performance.

**3.1 Formula and Approach**

To classify training examples, we calculated confidence scores, which measure the model's certainty in predicting the correct label for each example. Confidence for a data point is defined as the mean probability of the correct label across all epochs during training. Formally, the confidence is given by:

Using confidence scores, we classify examples as:

* Easy-to-learn: Examples consistently predicted correctly with high confidence ( > .7)
* Hard-to-learn: Examples consistently predicted incorrectly or with low confidence( < .3)

By focusing on these metrics, we aim to identify and leverage the most impactful subsets of training data to improve the model's generalization capabilities.

**3.2 Dataset Modification**

After computing confidence scores for each example, the training dataset is modified by adjusting the proportions of easy-to-learn and hard-to-learn examples. This process begins by identifying examples based on predefined confidence thresholds. Examples with high confidence scores are classified as easy-to-learn, while those with low confidence scores are labeled as hard-to-learn. Using this classification, we construct modified training datasets with varying proportions of these subsets, ranging from 10% to 90% hard examples. Importantly, the overall size of the dataset remains constant to ensure a fair comparison across experiments. This systematic adjustment of dataset composition allows us to investigate how the presence of easy-to-learn and hard-to-learn examples affects the model's ability to generalize and perform on unseen data.

**3.3 Experiment Settings**

To evaluate the impact of dataset composition on model performance, we conduct experiments with the following settings. First, a baseline model is trained using the full, unmodified SNLI training set to establish a reference point for comparison. Then, the model is trained on modified datasets with different proportions of easy-to-learn and hard-to-learn examples. These variations include extreme cases, such as training with 90% hard examples, as well as balanced cases, like a 50-50 split between easy and hard examples. Finally, the model's performance is evaluated on the full SNLI validation set using standard metrics like accuracy. Additionally, we measure performance on specific validation subsets, such as only hard examples, to assess the direct impact of training with challenging data. By systematically varying the dataset composition and comparing results across these configurations, this approach provides valuable insights into how the composition of training data influences model learning and generalization.

Contrastive Training

Guided by “Learning with Instance Bundles for Reading Comprehension” (Dua et al., 2021), we took the approach of contrastive training with the assumption that while in most training datasets, the samples are all identical and independently distributed. However, there are some cases, for example, when using samples that are slightly or subtly modified, we cannot use the assumption that training is completely i.i.d. In our case, we can see that our model is struggling with subtle distinctions, especially within the neutral class space. In this case, we propose contrastive training  that will help structuring the embedding space such that similar pairs are closer and vice versal, which will help the model understand general language semantics.

**4.1 Data Pairs and Preprocessing**

We first start by identifying the premise-hypothesis pairs that have labels for entailments, neutrals, and contradictions. Therefore, we will have positive, negative, and neutral pairs. We then use our Electra-small pretrained model, and we generate a triplet amongst these bundles. In these triplets we include the following:

* Anchor: the premise sentence
* Positive: hypothesis sentence with the positive label
* Negative: hypothesis sentence with a negative label

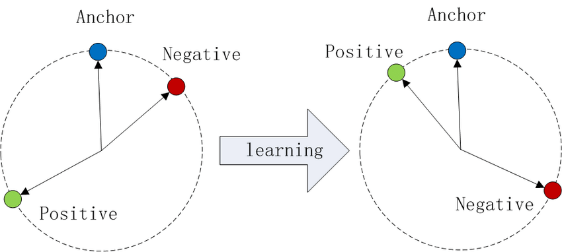
Now, this code is ready for training, as we are allowing the model to have a premise associated with a positive statement and a negative statement. This will allow the model to learn both positive and negative labels associated with the premise, giving it a good understanding of both perspectives.

**4.1 Triplet Loss**

After preprocessing the code, we determine our loss function, which for contrast training is given by the triplet loss function:



where the loss function enforces a margin (a) between positive and negative pairs that are relative or close to the anchor. This loss function encourages embeddings to be created for similar pairs while penalizes those that are dissimilar. Another representation of this is shown through



Wikipedia - Triplet Loss

which shows the impact of training that appreciates positive examples and penalizes negative examples, making the model understand that positive examples are closer in topic, context and score to the anchor, and negatives are not. We then combine the contrastive loss with the general classification loss of the 3 labels and fine-tune the model using a combined loss methodology.

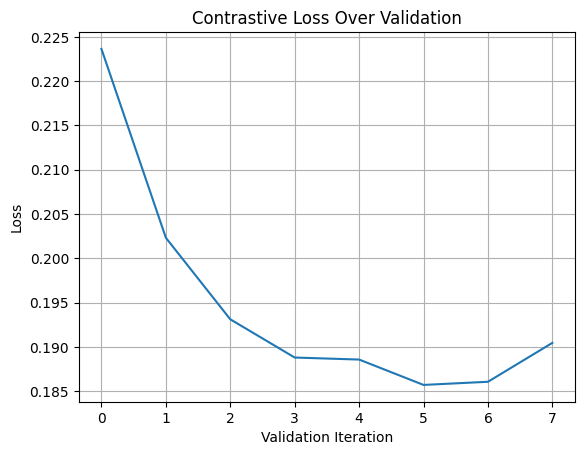
Results

When examining the results of the model and being able to measure if we truly made an impact, we looked at several performance metrics, ranging from confusion matrices, to specific losses according to our approaches.

For Dataset Cartography, we took the approach of \_\_\_\_\_\_\_\_\_\_\_

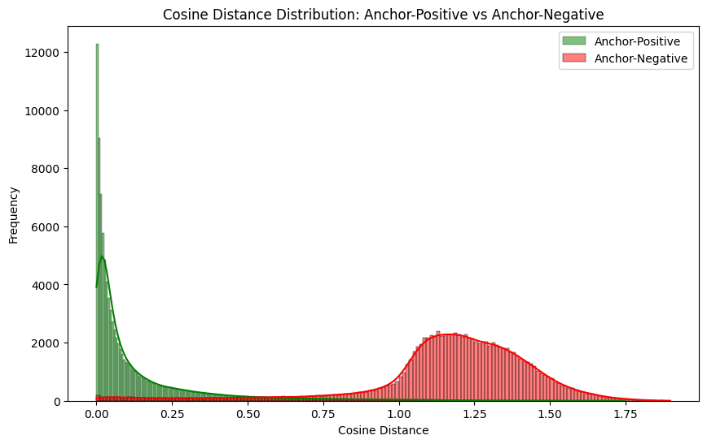
**5.1 Contrastive Training Results**

For the Contrastive Training approach, we created the loss function as described in 4.1 Triplet Loss and trained the model to try to minimize that loss function as much as possible. Looking back at the fundamental purpose of the loss function, we can see that it attempts to increase the distance between the premise or anchor, and the negative examples, and therefore penalizes the model every single time it thinks of a negative example as “close” to the anchor. For positive examples, it tries to encourage closer and closer examples. This triplets loss, allows the model to understand the positive and negative that can possibly be correlated with a given anchor, allowing it to detect extreme cases. Our hypothesis was, that if we are able to give our model the ability to train based on positive and negative examples at the same time, which shows dependencies, then we can also allow for the model to better establish neutral statements, which will be scored as a midpoint between positive and negative examples. With this hypothesis, we saw the minimization of the triplet loss on our validation sets.

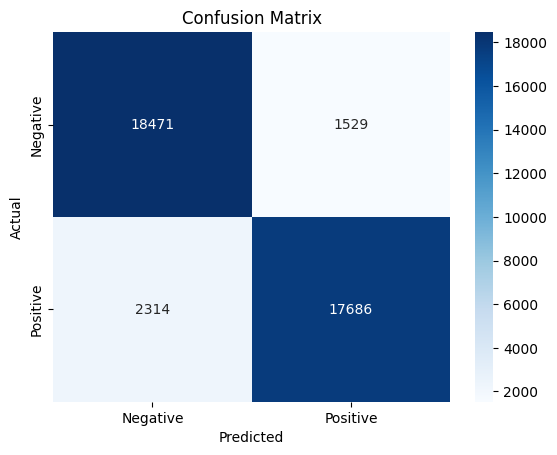


As we can see, the loss function maximized at the 5th epoch, where it was around .185. However, studies show that any triplet loss that is less than .20 is good evidence that the model is doing a good job in identifying positive and negative examples the way it should. This gave us a good idea that our model was performing the way we were expecting to, and training properly with every epoch.

We then took a deeper look at the actual cosine equation, or in order words, the triplet loss, to see the distance that was being observed, and decided for training and evaluation by the model.



The cosine distance allows us to see how the model perceives the example, and we get to see through this graph that most positive examples have a distance of close to 0 and less than 0.5, whereas all the negative examples in the validation set have a distance of greater than 1.0. This representation shows the minimization of loss in this function, as well as correct addressing of both sentiments of examples. Lastly, our final observation from this specific model was the confusion matrix shown below



Through this example, we see how the misclassification of examples was less than 10% of the time. This indicated a strong model performance, and an improvement from the baseline model.

**4.2 Overall Review**

Retrospectively, we can see that our modifications helped improve our model and fix the issues that we had addressed in the initial sections. Below, we can see the final metrics of all 3 models together, where the enhancements of contrastive training and dataset cartography, led to a more-intuitive model with better semantics on English grammar, and better understandings of sentiments.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Loss |
| Baseline | .86 | .38 |
| Dataset Cartography |  |  |
| Contrastive Training | .91 | .16 |

Conclusion

We started this project with the intention of finding the issues and shortcomings of the Baseline Electra-Small model. Through this, we found a lot of issues, including issues in neutrality detection, and contradiction detection.

To address these fixes, we took two different approaches, using Dataset Cartography, which allowed the model to separate easy learners and hard learners, and detect the key sensitivities between both. Through Contrastive Training, we helped the model learn how to detect entailments better, as well as contradictions better, which by default helped in detecting neutrality as well. Overall, both our approaches made significant improvements to the model, and we were able to accomplish a better, more detail-oriented performance out of it.