Modeling Ambiguity in Natural Language Inference

Natural Language Processing (DSC-395T), Final Project, Fall 2022

Abstract

Natural Language is rife with ambiguity. Given this and the fact that Natural Language Inference (NLI) datasets are based on Natural Language, it is no surpise that these datasets contain ambiguity. When models are trained on ambiguous data and not given an opportunity to express uncertainty, one can quite naturally expect that these models will not perform well on ambiguous examples.

In this paper, I will explore ambiguity in the SNLI dataset (Bowman et al., 2015), analyze how this ambiguity affects model performance, propose an alternative approach to training models that takes into account and targets ambiguitity, train an Electra-small model using said approach, and analyze the results, hoping to improve model performance on ambiguous examples while minimizing performance degradation on unambiguous examples.

Much of this paper is based on ideas and approaches in the papers "Embracing Ambiguity: Shifting the Training Target of NLI Models." (Meissner et al., 2021) and "What Can We Learn from Collective Human Opinions on Natural Language Inference Data? (Nie et al., 2020).

1 Introduction

Even the most straightforward Natural Language is open to some interpretation. Any individual asked to classify a given piece of text is going to bring their own biases to the task. What happens when the problem is compounded by the fact that the meaning of the text is not clear? Maybe there are typos, misspellings, or different syntactic or semantic interpretations? How might one expect different individuals to interpret this?

If this is the case - that there is some inherent level of bias and ambiguitity in any Natural Language dataset - what is the best measurement of performance? If an individual was asked to classify a dataset of highly ambiguous examples and they scored poorly based on the gold labels, would they be deemed to not know or understand language well?

Maybe a better measure of a model's performance on interpreting Natural Language would be to allow the model to express its answers on a spectrum? For example, the SNLI dataset is assigned a single gold label of entailment, neutral, or contradiction for each given premise/hypothesis example. But for the dev and test datasets, multiple annotators were used to classify the examples. From these different annotations, we can quite reasonably infer (although not without other issues arising) ambiguity measurement: how ambiguous is the example (i.e., how many annotators aggreed on the most common label?) and what is the distribution of annotations (i.e., how many annotators classified the example with a given label)? And if we are going to hold a model's performance to a different standard (i.e., based on a distribution instead of discrete values), how might we modify the model's training to improve its performance?

In this experiment, I will train an Electra-small model on the SNLI dataset. As a baseline I will train the model using gold labels (i.e., the single, discrete values) and measure both its accuracy and the similarity between the predicted score/probability distribution and the annotator labels as a distribution (describe in detail below). I will then train another Electra-small model on the SNLI dataset, first training the model on the train dataset using gold labels, as with the baseline model (since this is the only label available), followed by fine-tuning the model on a modified dev dataset with the label as a probability distribution of the annotation labels.

The hypothesis is that the experimental model, when fine-tuned on the probability distributions, will outperform the baseline model on ambiguous data (defined below) when the performance is measured base on the similarity between the predicted

Dataset	Examples	Annotators
Train	550,152	1
Dev	10,000	5
Test	10,000	5

Table 1: SNLI Dataset Metadata

and annotation distributions, and that the more ambiguous the data (again defined below) the greater the difference will be in model performance.

All of the code for this paper is released here[1], a fork of this existing repository[2] with minor changes.

1.1 Data

The data used throughout this experiment is the SNLI dataset (Bowman et al., 2015), which, as can be seen in Table 1, consists of 550,152 training examples, 10,000 dev examples, and 10,000 test examples. Each example, for the purposes of this experiment, consists of a premise, a hypothesis, and a gold label from one of entailement, neutral, and contradiction. While the training dataset derives its gold label from a single annotation from the original annotator, the dev and test datasets derive their gold label from the concensus label from five different annotators (the original annotator plus 4 mechanical turk workers), with ties receiving a gold label of ’−’ (i.e., inconclusive).

1.1.1 Ambiguous examples

In the SNLI datasets, or any Natural Language dataset for that matter, ambiguity may come from many different sources:

Some ambiguous examples may be due to typos. For example, in the following example the text switches from *wandering* to *wonder*, which may or may not picked up on by all the annotators:
Premise: *A man wandering in the desert as the clouds roll in.*
Hypothesis: *A man and a camel wonder in the desert.*

While some ambiguous examples may be due to annotator interpretation. For example, in the following example, if one imagines a solid door, then the door must be open for the child to look out (i.e., entailment). But if one imagines a screen door, then the door doesn’t necessarily need to be open (i.e., neutral):
Premise: *A child is looking out of a door.*
Hypothesis: *The door is open.*

Or still some ambiguous examples may be due to different parsing. For example, in the following example, if one reads this as "two players ... one of which ...", then this would be entailment. But if one reads this as "two players ... and one other ...", then this would be contradiction:
Premise: *Two players are on a wet field and one is on the ground.*
Hypothesis: *There are only two people in the field.*

1.2 Model

The model used for this experiment is a Huggingface pre-trained Electra-small-discriminator model[3] with default configuration and hyperparameters.

2 Experiment Setup

2.1 Data Alteration

While the SNLI train dataset includes only a single annotator label, the dev and test datasets include five annotations for each example, making it possible to derive a proxy probability distributions based on the labels from the different annotators. While far from perfect (and certainly a source of bias), it is reasonable approach with very little additional effort. As an example for illustrative purposes, given the annotations [entailment, neutral, contradiction, entailment, entailment], we can assign the probability distribution [$\frac{\#\ entaiment}{5}$, $\frac{\#\ neutral}{5}$, $\frac{\#\ contradiction}{5}$], or [$\frac{3}{5}$, $\frac{1}{5}$, $\frac{1}{5}$] in this example.

For analysis purposes, we can also use the annotations as a proxy for assigning different levels of ambiguity for each example:

- If all five annotations are the same, we assign ’No Ambiguity’.

- If four annotations are the same, we assign ’Medium Ambiguity’.

- If three annotations are the same, we assign ’High Ambiguity’.

[1] https://github.com/rgross-utexas/fp-dataset-artifacts

[2] https://github.com/gregdurrett/fp-dataset-artifacts

[3] https://huggingface.co/google/electra-small-discriminator

Dataset	Total	None	Med.	High	Inv.
Dev	10,000	5,479	2,849	1,514	158
Test	10,000	5,368	2,874	1,582	176

Table 2: Ambiguity distribution for the SNLI dev and test datasets. Those marked Inv. do not have a single mode and are therefore removed from the datasets prior to training.

Training	Amb. Level	Acc, ↑	JSD ↓
Baseline	All	0.8890	**0.1902**
	None	**0.9544**	**0.1045**
	Med.	0.8847	0.2621
	High	0.6745	0.3510
Uniform	All	N/A	0.4832
	None	N/A	0.5641
	Med.	N/A	0.4164
	High	N/A	0.3290
Experiment	All	**0.8908**	0.2119
	None	0.9539	0.1852
	Med.	**0.8858**	**0.2233**
	High	**0.6852**	**0.2823**

Table 3: Main results from the experiment: The first group of data show the baseline performance, with the model trained on the train dataset with gold labels, fine-tuned on the dev dataset with gold labels, and evaluated on the test dataset with gold labels. The second group shows the performance given a uniform distribution, to compare a model that always predicts an equal likelihood for each label. The third group of data show the experimental model performance, with the model trained on the train dataset with gold labels, fine-tuned on the dev dataset with probability distributions, and evaluated on the test dataset with probability distributions.

- If two annotations are the same, we discard the example as there is no single mode.

This grouping will be useful in showing that there is a strong negative correlation between model performance and the level of ambiguitity of an example. But, more importantly for this experiment, we will be able to use this grouping to evaluate the second part of the hypthesis (i.e., that the more ambiguous the data the greater the difference will be in model performance).

Therefore, for each example in the SNLI dev and test datasets, the *label* column was replaced based on the approach above.

2.2 Model Alteration

In order to train the model on the probability distributions, a custom `ElectraForSequenceClassification`[4] model with a soft cross entropy loss function was created, following the work done in the AmbiNLI repository.[5]

3 Baseline Analysis

3.1 Model Training

For a baseline, we trained the Electra model with the SNLI train dataset using gold labels (there are only gold labels for this dataset), fine-tuned the model with the SNLI dev dataset using gold labels, and evaluating the model on the SNLI test dataset using gold labels.

3.2 Results

As we can see from Table 3, the baseline model has an overal accuracy of 88.90%, while, as expected, the accuracy decreases as the level of ambiguitity increases. For the Jensen-Shannon divergence, which is the measurement we are interested in, the baseline has an overall score of 0.1902, while again, as expected, the score increases as the ambiguitity level increases. It is worth pointing out that the baseline score for highly ambiguous examples (0.3510) is worse than the score for a model using a uniform distribution for the same examples (0.3290).

This is mainly due to the fact that when a model is trained on gold labels alone, it learns to put all its eggs in one basket, creating heavily skewed predicted probability distributions.

Even if the prediction is correct from a gold label perspective, its predicted probability distribution will be highly dissimilar from the annotation based probability distribution. For example, in one example where the gold label was correctly predicted (i.e., neutral), the annotation distribution of $[0.4, 0.6, 0.0]$ and predicted probability of $[0.0043, 0.9816, 0.0141]$ lead to a Jensen-Shannon divergence score of 0.3954.

And when the prediction is incorrect, the distribution dissimilarity can be quite extreme. For example, in one example where the gold label was incorrectly predicted, the annotation distribution of $[0.4, 0.6, 0.0]$ and the predicted probability of $[0.2814, 0.07764, 0.6409]$ lead to a Jensen-

[4] https://github.com/rgross-utexas/fp-dataset-artifacts/blob/final-project-work/custom_electra.py

[5] https://github.com/mariomeissner/AmbiNLI/blob/main/scripts/custom_bert.py

Shannon divergence score of 0.5844. One can see that even though the model was able to express a level of uncertainty, the diverence score is still quite high.

4 Exeperiment Analysis

The goal of the experiment is to show that when fine-tuned on the annotation distributions derived from five annotators, the model will outperform the baseline model on ambiguous data and that the more ambiguous the data, the greater the difference will be.

4.1 Model Training

For the experiment, we trained the Electra model with the SNLI train dataset using gold labels (there are only gold labels for this dataset), fine-tuned the model with the SNLI dev dataset using the derived annotation distributions, and evaluating the model on the SNLI test dataset using the derived annotation distributions.

4.2 Results

As we can again see in Table 3, the experimental model, while performing nearly identically, if not better, to the baseline model in terms of accuracy, it obviously outperforms the baseline model on the medium and highly ambiguous data groups when it comes to the Jensen-Shannon divergence.

It is quite satisfying to see that by making a relatively simple change to a readily available dataset, we can make such definitive improvements on the model performance on ambiguous data.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Johannes Mario Meissner, Napat Thumwanit, Saku Sugawara, and Akiko Aizawa. 2021. Embracing ambiguity: Shifting the training target of NLI models. *CoRR*, abs/2106.03020.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.