

Inference Detection in NLP Using the MultiNLI Dataset

Term Paper, CSC 820, Spring 2022

Chris Huber*
SFSU

ABSTRACT

This paper describes the process of classifying inference from the MultiNLI dataset, a corpus of over 422,000 sets of sentences. Inference involves examining pairs of sentences and determining if they are a contradiction, an entailment, or neutral. There are several models capable of this including RoBERTa and XLNet which can be fine-tuned to produce increasingly better accuracy.

1 INTRODUCTION

Natural language inference revolves around determining whether a hypothesis is an entailment, contradiction, or neutral. An example is shown in Table 1. The MultiNLI corpus has 0.9 and 1.0 versions, both of which are examined (TBD).

To perform this task, I established a baseline accuracy using an algorithm called RoBERTa which was developed by Google in conjunction with Fairseq¹, a Google-developed open-source toolkit designed for translation, data modeling, and other text related tasks.

RoBERTa is based on BERT (Bidirectional Encoder Representations from Encoders) which was developed by the Google AI team and is discussed in a paper called "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding".² In a nutshell, it works by using a Masked Language Model (MLM) which hides some of the tokens in the input and attempts to derive the missing words using the surrounding context. This also works for next-sentence prediction which is the use case for inference.

There are three baseline neural network implementations for RoBERTa which are based on CBOW (Continuous Bag of Words), bi-directional LSTM (Long Short-Term Memory), and ESIM (Enhanced Sequential Inference Model).³ These are implementable by running the train_mlni.py script which uses TensorFlow to train the model on either MLNI data, a mix of MLNI and SLNI data, or on a single genre in the MLNI dataset. Dropout is used in all three implementations for regularization.

The inspiration for this project came from a list of tasks posted to the SuperGLUE website.⁴ (The GLUE (General Language Understanding Evaluation) benchmark is a benchmark set of NLP tasks to be performed on sentence pairs which has an ongoing competition ranked by accuracy). RLE (Recognizing Textual Entailment) is a very current topic in NLP technology research and I was inspired to see what I could contribute. There have also been Kaggle competitions involving inference determination.⁵

I propose, after getting benchmark results for one or more of the datasets, to refine the algorithm and/or try other methodologies in an effort to improve my F1 score.

*email:chrish@sfsu.edu

Premise	Hypothesis	Label
A man inspects the uniform of a figure in some East Asian country	The man is sleeping.	contradiction
An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor	neutral
A soccer game with multiple males playing	Some men are playing a sport	entailment

Table 1:

Example of evaluations of sentence pairs. Taken from http://nlpprogress.com/english/natural_language_inference.html

2 RELATED WORK

Sample citation to keep Latex happy [1].

A paper entitled "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference" by Adina Williams, Nikita Nangia, and Samuel Bowman⁶ discusses the creation of the MultiNLI (Multi-Genre Natural Language Inference) corpus. It is comprised of 433,000 annotated examples designed to train machine learning algorithms. It offers data from 10 distinct genres of written and transcribed oral communication.

Prior to MLNI, the Stanford NLI corpus was the largest available corpus but fell short in a couple of ways. First, it was all from a single genre due to the fact that it was drawn only from image captions which lacked the robust language varieties that exist. The authors found it insufficient to provide a good benchmark for NLU (Natural Language Understanding).

RoBERTa is a technology developed by Google described in the paper "RoBERTa: A Robustly Optimized BERT Pretraining Approach" by ⁷.

An article entitled "Transformers: Retraining roberta-base using the RoBERTa pre-training procedure" ⁸ details the process of re-training RoBERTa on a custom dataset. For reference, it mentions that the Roberta-Base was "trained on 1024 V100 GPUs for 500K steps." It suggests using the TensorFlow transformers library to retrain a RoBERTa neural network with new data, which is the process I will follow.

3 IMPLEMENTATION

I started by evaluating the data in the MLNI 0.9 dataset for the Kaggle competition, since that is what they used. It consists of 9796 sentence pairs and has a fairly even distribution of genres as shown in Table 2. It is comprised primarily of sentences under 200

¹<https://aclanthology.org/N19-4009.pdf>

²<https://arxiv.org/pdf/1810.04805.pdf>

³<https://github.com/NYU-MLL/multiNLI>

⁴<https://super.gluebenchmark.com/tasks/>

⁵<https://www.kaggle.com/c/multiNLI-matched-open-evaluation/data>

⁶<https://cims.nyu.edu/~showman/multiNLI/paper.pdf>

⁷<https://arxiv.org/pdf/1907.11692.pdf>

⁸<https://towardsdatascience.com/transformers-retraining-roberta-base-using-the-roberta-mlm-procedure-7422160d5764>

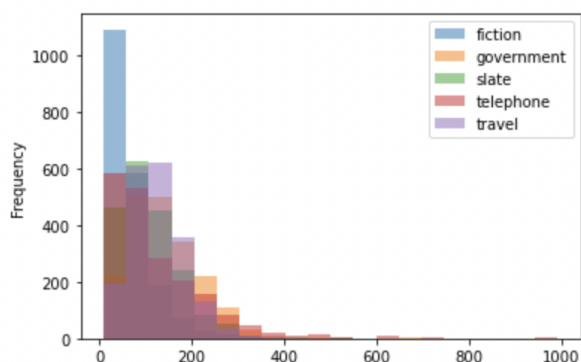


Figure 1: Distribution of sentence lengths by genre in the Kaggle competition dataset. (Taken from <https://www.kaggle.com/code/takahirokubo0/multinli-dataset-analysis/notebook>)

Genre	Count
Fiction	1978
Travel	1964
Telephone	1955
Government	1953
Slate	1946

Table 2:

Genre counts for Kaggle dataset.

characters with a few outliers with a very long length as shown in Figure 1.

I used the PyTorch framework to evaluate the data using tensors which are trained to predict

$$y = \sin(x) \text{ from } -\pi \text{ to } \pi.$$

by minimizing squared Euclidean distance. Tensors are similar to NumPy arrays but can be run on either the CPU or GPU. I used the PyTorch `no_grad()` option to disable gradient calculation data which is otherwise tracked for later calculations to optimize the code as shown in Listing 1.

```

1 with torch.no_grad():
2     for k in range(len(test_s1)):
3         # Encode a pair of sentences and make a
          prediction
4         tokens = roberta.encode(test_s1[k], test_s2[k])
5         prediction = roberta.predict('mnli', tokens).
          argmax().item()

```

Listing 1: Code to perform predictions using RoBERTa and PyTorch.

Another ongoing competition is hosted by SuperGLUE for which entailment is one competition topic of ten total. The data is defined in a similar fashion with headings titled "premise" and "hypothesis" and a corresponding index. However, the competition only accepts two labels for entailment and non-entailment whereas the Kaggle competition required labels for contradiction, neutral, or entailment. To further illuminate the baseline performance for RoBERTa, I submitted its analysis of the dataset to the competition.

4 EVALUATION

I configured and ran the RoBERTa algorithm against the MLNI corpus to establish a baseline. It performed quite well, resulting in a

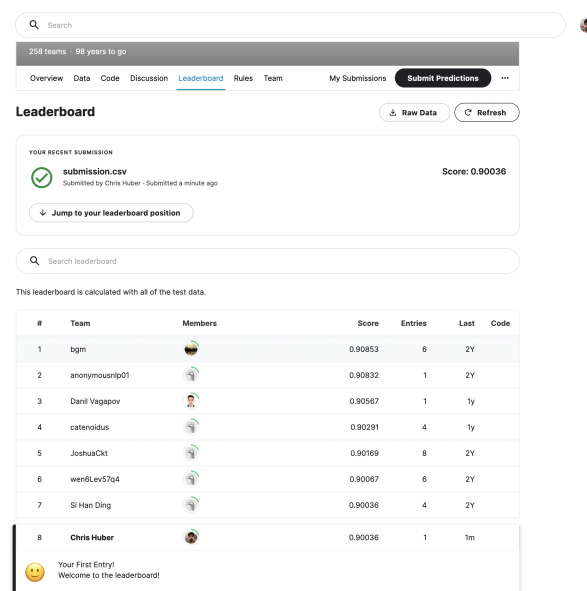


Figure 2: Results from first run of RoBERTa against the MultiNLI dataset.

score of .90006 which ranked as a tie for 7th out of 263 submissions. This model was pre-trained on the MLNI dataset which is available as an open-source download.

I proceeded to attempt to train the model on just the train set provided by the competition to see if it would increase the ranking.

ACKNOWLEDGMENTS

The author wishes to thank Professor Sam Bowman from NYU whose team curates the MNLI corpus .

REFERENCES

- [1] J. L. Hennessy and D. A. Patterson. *Computer Architecture, Fifth Edition: A Quantitative Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th ed., 2011.