Text Entailment With Switched Input Sentences

Vamsi Sistla, Terence Davis
University of California at Berkeley; Berkeley, CA
{vsistla, lisnter}@berkeley.edu

Abstract

Pretrained language models are becoming commonplace in a wide variety of applications. Although these systems are highly accurate there is room for significant improvement in tasks such as text entailment (TE) in real-world situations. For example, we want to make sure these models perform well when dealing with non-native English speakers who might use unusual constructions or place their sentences out-of-order. In this research, we have measured the overall performance of RoBERTa ([1]) when sentence pairs are switched (sentence 1 -> sentence 2 and sentence 2 -> sentence 1) while retaining the same label and have further analyzed the model's response where the originally assigned label was *contradiction*. We find that RoBERTa doesn't perform well in certain types of sentence pairs (that are contradictory) due to linguistic complexities (ambiguity for example) present in the English language. We are particularly focused on those switched labels that predict entailment from contradiction. We consider label switches in cases of mission critical applications such as health care, emergency, security, privacy and terrorism where the consequences of an incorrect or ambiguous prediction can be dire.

Our research highlights the importance of building language models that take into account linguistic complexities and provide future recommendations on model improvements with respect to those sentence pairs where the label changed. We discuss the importance of creating better benchmark datasets that include these linguistic ambiguities. Finally, we challenge the notion that English sentence pairs can only be categorized with three different labels - Neutral, Entailment and Contradiction.

1 Introduction

Our research focuses on the effectiveness of RoBERTa in addressing text entailment (TE) tasks for specific types of use cases and users. For example, we wanted to understand the performance of this model where it is utilized by a chatbot or IVR system that interacts with non-native English speakers or with children who are just learning to form sentences. Our assertion is that during these use-cases, the system must properly classify a wide range of TE scenarios where the input sentences do not conform to the relatively well-structured data upon which RoBERTa and BERT were trained on; e.g. poor or rare word-choice; awkward and/or reversed sentence ordering. In these situations, the system may not show consistent results across the three allowed labels: (1) Entailment, (2) Neutral and (3) Contradiction. Our research does broad

analysis on all three labels, but we are specifically interested in contradictory sentences. We want to understand situations where an assignment of contradiction is invariant to sentence order and poor word choice.

We start with the RoBERTa base model and pretrain using WikiText-103 ([3]) and finetune using MNLI data. This prepares us to get a base-line accuracy of TE against the 3 labels (entailment, neutral, contradiction). We then move on to the key question of our research; that of using the GLUE benchmark data ([4]) MNLI task where sentence 1 and sentence 2 are switched during the prediction phase to ascertain if the assigned label remains the same or changes. We then move on to repeat this testing regimen against RoBERTA.large.mnli ([1]). More details of this assessment are provided in Section 3 of this paper.

2 Background

With the release in 2018 of the Bidirectional Encoder Representations from Transformers, or BERT ([2]), a revolution in self-supervised pre-training began. RoBERTa has built upon BERTs success by altering key elements of BERT and has emerged as a highly accurate language model[1] since its release in 2019. Since that publication of that work, much additional refinement has been done to improve on RoBERTa's base-line performance through the modification of various model pre-training hyper-parameters. The richness of the underlying models and newness of the architectures affords many such opportunities.

The current research utilizes the RoBERTa.base and Roberta.large.mnli models

3 Experimental Setup And Evaluation

3.1 Implementation

We have used the Fairseq Sequence Modeling Toolkit ([7]) to train and score our fine-tuned models. All code uses PyTorch ([8]), Python and standard bash scripts. The majority of our work was executed on the TrainML.ai ([9]) GPU infrastructure however we've also leveraged Google GPU and TPU instances. To reduce pretrain and fine-tuning durations we have installed NVIDIA's apex ([10]) library on our GPU machines. The custom model was trained on a 4 GPU instance within the TrainML.ai infrastructure.

Roberta.base was fine-tuned using the full Wikitext-103 dataset ([3]). After fine-tuning we tested the model against the MNLI matched dataset and obtained an 87.4% accuracy which is on-par with the measurements obtained from Liu ([1]) when using their more accurate RoBERTa.large model.

We have pretrained RoBERTa base model with our custom dataset we generated from the Wikitext-103 corpus.

We have conducted over twenty five experiments varying the hyperparameters as indicated in table 1 (in Appendix). These experiments used the WikiText-103 dataset and the Roberta.base model.

During the fine tuning phase of our custom model, we have used fine tuning parameters listed in Table 2 (in Appendix) to train the pretrained model using GLUE MNLI dataset([5]).

3.2 Data

Along with WikiText-103 for pretraining, we finetuned using the MNLI dataset. The

Multi-Genre Natural Language Inference (MNLI) corpus was parsed into multiple subsets. MNLI is commonly used for NLP operations requiring sentence-level analysis. For this work, we focused on subsets that provided insight into linguistic patterns afforded by the deep variety of source material. For this effort, all text operations began with the JSON files.

All files were prepared for input to the pre-training by first being passed through a Python-based parser that created the output files. The source JSON file is directly prepared for consumption by the RoBERTa pre-training scripts; in particular this is one set of *test, train and valid* files for each of the input fields (sentence1, sentence2 and gold_label).

Once correctly processed, the input files are prepared for inference (scoring) in two stages. The first stage generates a byte pair encoding vocabulary file (.bpe) which is provided as input to the second stage; the second stage builds the vocabulary and binarized training data. The flow is depicted in figure 2 (in the appendix). In a similar fashion we trained using SNLI dataset ([6]) however due to poor performance, we are not reporting those results here.

3.3 Evaluation using GLUE dataset

After validating the data we performed the research experiment by testing the accuracy of the model against the switched validation set (i.e. sentence 1 and sentence 2 swapped position but with the original label).

The following table (Table 3) shows the performance of two models - one represents our custom developed model with the other from the benchmark against switched GLUE provided input (originally their "dev matched" dataset).

Model	GLUE TE Benchmark for Dev Matched Dataset	Switched GLUE Dev Matched Dataset
Roberta.base Wikitext-103 Pretrained, GLUE MNLI Finetuned	87.40%	**55.34%**
Roberta.Large.MNLI	90.20%	**59.12%**

Table 3 - Model Performance between benchmark and switched datasets

Table 4 below shows the results of Roberta.large.mnli's scoring of the GLUE Dev Matched dataset when Sentence 1 and 2 are switched (sentence 1 and sentence 2 switch positions).

Model	Switched GLUE dev matched dataset		
	Entailment %	**Neutral %**	**Contradiction %**
Roberta.Large.MNLI	32.36%	73.38%	78.56%
	67.64%	26.62%	21.44%

Table 4 - Model stability of the prediction of each of the labels

These results show that almost 60% of entailments were assigned an incorrect prediction from the switched data set while less than 30% of neutral and contradiction labels show a similar prediction instability. Linguistically speaking, the entailment label is directional in nature and thus it is natural for the predicted label to change when the sentences are switched; however neutral and contradiction cannot be considered as directional in every scenario. For example, if a pair of sentences are neutral, then they should also be neutral when you switch the order unless they exhibit some uncommon quirk of the English language. Our research focuses on contradiction performance which exhibits some interesting behaviors.

We continue to evaluate (in Table 5) why the 22% of contradiction sentences switched their predicted label. Out of the 3185 sentence pairs with gold label assignment of *contradiction*, less than 2% of them (38 pairs) switched to entailment. Pairs with differing labels going from *contradiction -> entailment* are more linguistically and algorithmically interesting than those going from *contradiction - > neutral*. To put this in concrete terms, if a chat bot encounters a contradiction->neutral pair, it could potentially respond back asking for clarification whereas it might not do the same with a contradiction->entailment pair.

Gold vs Switched Prediction		
	Count	%
Contra / Contra	2502	78.56%
Contra / Neutral	645	20.25%
Contra / Entailment	38	1.19%
Total Contradictions	**3185**	

Table 5 - Contraction Gold Labels - Switched Prediction Percentages

An event that occurs 2% of the time when applied to Internet-scale applications could occur many millions of times per day and dozens of times per day in mission critical applications such as health care, security, privacy and finances. Thus the case for improved accuracy is clear.

	Contradiction Error Breakdown	Breakdown Reason %
	Complex Sentences	15.79%
	Gold label misclassification	23.68%
Language Complexities	Ambiguity in Language	26.32%
	Sentences With Quantization	7.89%
	Sentences with Question	15.79%
	Sentences measuring time	5.26%
	Sentences that have a typo	5.26%

Table 6 - Contradiction Error Breakdown

When we further dissect table 6 above and ask why these contradictions are switched, we find that over 55% of them are due to language complexities, such as language ambiguity, sentence quantization, sentences with questions and sentences with time or position. Over 22% of them were also caused by incorrect labeling of the gold label.

3.4 Only Three labels for TE?

We fundamentally challenge the notion of only three labels - Neutral, Entailment and Contradiction - when it comes to the text entailment (TE) task. We attribute some of the performance challenges we have observed in the models to the generalization of English sentence pairs into just these three label categories. On top of that, neutral itself is a very broad category that could be split into finer categories such as irrelevant, related but not directional, etc. This type of expansion in the public datasets and models will invariably lead to new and different challenges in building and training

models but will be a worthwhile improvement.

4 Future Possibilities

We feel TE task datasets (GLUE dev matched) are littered with many mislabeled records which has certainly confused the leading models when dealing with our switched sentence dataset. We should continue to update some of the benchmark data so that models can fine tune based on a more complete corpus (from a human and linguistic perspective).

We also took notice that model performance decreases when faced with certain types of sentences, i.e. those containing linguistic complexities such as ambiguity, positional details (top/bottom, up/down, front/back, etc.), complex sentences and so forth. In order to create better NLP models, we need to create more and more varied datasets that are representative of some of these linguistic complexities.

5 Acknowledgements

The authors deeply thank Adam Kowalski at TrainML.ai who graciously granted us 500 GPU hours on his platform to enable 4-GPU training capability.

6 References

1. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arXiv preprint arXiv:1907.11692*.
2. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805
3. Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016
4. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy and Samuel R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. 2018. From *ACL Anthology* at https://www.aclweb.org/anthology/W18-5446/
5. Adina Williams, Nikita Nangia and Samuel R. Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* at http://aclweb.org/anthology/N18-1101
6. 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)* at http://nlp.stanford.edu/pubs/snli_paper.pdf
7. Fairseq: A Fast, Extensible Toolkit for Sequence Modeling
Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, Michael Auli

8. Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In NeurIPS Autodiff Workshop, 2017.
9. Trainml.ai – http://www.trainml.ai
10. NVIDIA.apex - https://github.com/NVIDIA/apex

7 Appendix

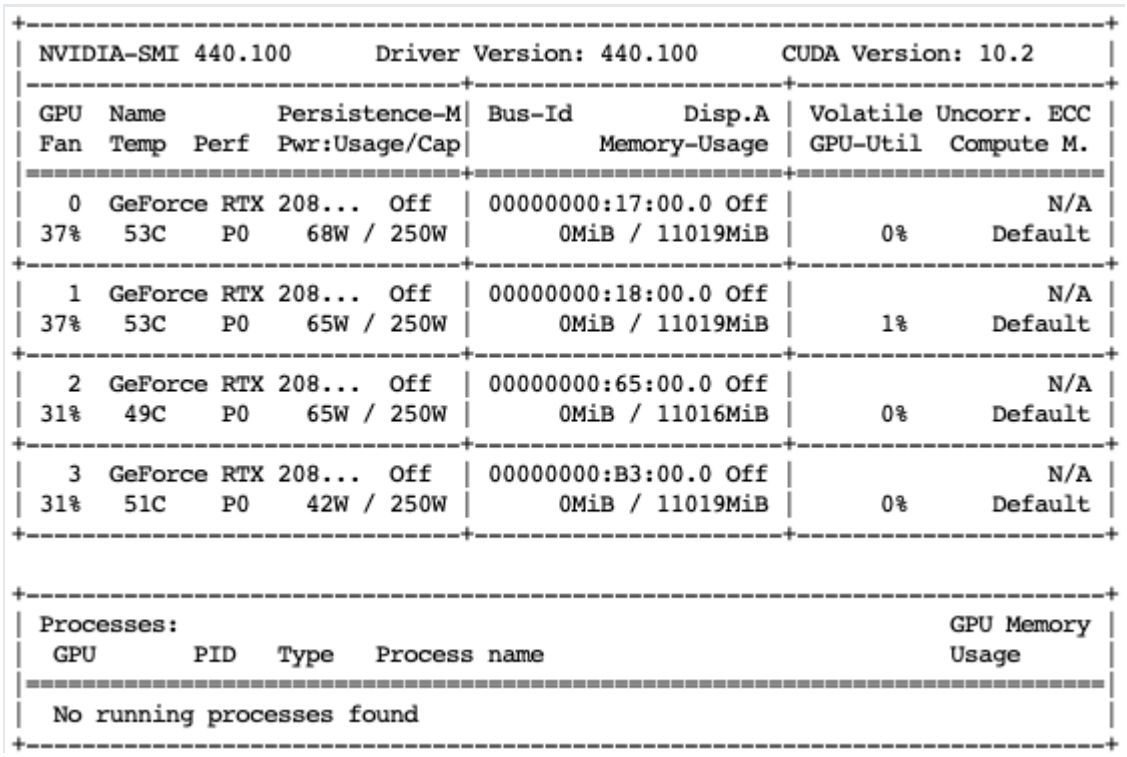
Table 1 Hyperparameters for RoBERTa Base Pretraining

Parameters with final values	Definitions of Parameters	Range of values
TOTAL_UPDATES=125000	Total number of training steps	10000 to 125000
WARMUP_UPDATES=10000	Warmup the learning rate over this many updates	100 to 10000
PEAK_LR=0.000003	0.0003 Peak learning rate, adjust as needed	1e-03 to 3e-06
TOKENS_PER_SAMPLE=512	Max sequence length	512
MAX_POSITIONS=512	Num. positional embeddings (usually same as above)	512
MAX_SENTENCES=12	Number of sequences per batch (batch size)	2 to 20
UPDATE_FREQ=64	Increase the batch size 16x	2 to 64

Table 2 Fine Tuning Parameters of Pretrained Roberta Base Model

Parameters with final values	Definitions of Parameters	Range of values
TOTAL_NUM_UPDATES=2036	Total number of training steps	2000 to 4000
WARMUP_UPDATES=122	Warmup the learning rate over this many updates	100 to 500
LR=1e-06	0.0003 Peak learning rate, adjust as needed	1e-03 to 3e-06
NUM_CLASSES=3	Number of classes for TE - Neutral, Entailment & Contradiction	
MAX_SENTENCES=12	Batch size	4 to 64
UPDATE_FREQ=64	Update frequency	2 to 64
CUDA_VISIBLE_DEVICES=0,1,2,3	Number of GPU Machines Used	1 to 4

Figure 1 Four GPU Machine Specifications



Some examples of Contradictory Sentence Pair Analysis

Here are some of the examples of error types of contradictory sentence pairs that switched during our experiment.

Error type	**Complex sentences**
Description	A complex sentence is one with multiple subordinate clauses or that are longer than the average.
Example	Sentence 2 - NHTSA concluded that **while section 330 superseded the section 32902 criteria**, **it did not supersede** the section 32902 mandate t**hat there be CAFE standards** for model year 1998.
Error type	**Gold label misclassification**
Description	Since the original gold label classification was done by human reviewers it is inevitable that errors creep into the final data. A few sentence pairs were thus mislabeled which would confuse the output.

Example	Sentence 1 - This is how things are and there are no apologies about it. Sentence 2 - Sorry but that's how it is. These two sentences are saying the same thing: this is how things are/that's how it is
Error type	**Ambiguity in language**
Description	Human languages are complex with many rules and almost as many exceptions to those rules. This is especially true of English and so it is not unexpected that ambiguous language constructions would be imperfectly represented in the model.
Example	Sentence 1 - She looked older even though she was only barely eighteen. Sentence 2 - She was quite young, not more than eighteen. This should still be a contradiction. There is no implied order so the model should have caught the contradiction. The phrase only barely eighteen means very slightly older than eighteen while not more than eighteen means up to seventeen years and 364 days. If the original predictions detected this subtly then the switched sentences should not have confounded the model either.
Error type	**Sentences With quantization**
Description	Teaching NLP systems about quantities and directions is notoriously difficult. Recent work has managed to convey the concepts of 1, 2, 3 and four but after that the models are unable to discern differences in specific quantities. Some of the corpus has sentences that depend on various forms of quantity, direction, etc. which are difficult for the model to handle.
Example	Sentence 1 - The only way to watch Washington Week in Review is from the start to the end, as anything else would be viewed. Sentence 2 - How to Watch Washington Week in Review : Back to front. These two sentences clearly are contradictory: "start to end" is the opposite of "Back to front" yet the experiment did not capture this difference while the original analysis did.

Error type	**Interrogative sentences**
Description	Within the errors caused by an imperfect model of English interrogative sentences were the largest separate category. Questions can be structured many different ways which can be difficult for the model to handle appropriately.
Example	Sentence 1 - You have to ask Severn about the four Jarvis children. Sentence 2 - The four Javis children? asked Severn. These two sentences unambiguously say to ask Severn about the Jarvis children. Perhaps the inversion of order in sentence 2 has thrown the predictor off.
Error type	**Sentences measuring time**
Description	Time is a very abstract concept and much like counting or quantities it is very difficult to model.
Example	Sentence 1 - The only time the disorder seemed to exist was before Ritalin came around. Sentence 2 - The disorder hardly seemed to exist before the stimulant Ritalin came along. These two sentences have a temporal and opposite relationship. Did the disorder exist before or after Ritalin? The concept of before vs. after was not captured properly during the prediction.
Error type	**Sentences with typo(s)**
Description	When attempting to model sentence entailment slightly different words can dramatically alter the intended meaning of the entire sentence and confuse the predictor.
Example	Sentence 1 - It is a matter of not having *nay* patients. Sentence 2 - It is really a matter of waiting. Sentence 1 in this example actually has a typo and a mis-spelling that happend to be a valid (and common) word which confused the predictor. In this situation, patients (as-in for a doctor) was likely meant to be *patience* (meaning an ability to wait). In the original analysis, it was properly identified as a contradiction however in this experiment it was mis-classified.

Figure 2 Preprocessing & Binarization of Input Data