EXPERIMENTAL PROTOCOL xCS224U APRIL 15 2020
Alan Dunetz

1. **Hypothesis**:

We plan to apply BERT to the NLI problem of determining whether a premise entails, contradicts or has no relation to a hypothesis.

We are particularly interested in comparing BERT's performance on premise/hypothesis pairs to a hypothesis-only baseline. [4] and [5] note that datasets where hypotheses are generated via crowdsourcing may contain artifacts or statistical irregularities that enable a model to classify a hypothesis without looking at the premise. We would like to explore BERT's ability to identify such artifacts.

We will also compare the performance of BERT to an LSTM model using GloVe vectors for both the premise/hypothesis case and the hypothesis-only case. Some questions we hope to explore include:

a) Can the use of contextual word embedding result in improved performance on the NLI task versus the simpler model?
b) Does contextual embedding assist in the identification of annotation artifacts and statistical irregularities?
c) Compared to the simpler model, how much better does BERT perform on the premise/hypothesis pairs dataset versus the hypothesis-only dataset, i.e., what percentage of BERT's performance is actually derived from the premises?
d) How does this performance vary across the different labels?

We hope we may also provide additional qualitative insight into the nature of artifacts/irregularities in crowdsourced hypotheses.

2. **Data**:

We plan to use the SNLI dataset which is available on the course repository.

We may also use the simple word pair premise hypothesis dataset that was provided for homework three.

We may also use the "hard" SNLI test set released in [5]. This may be reserved as our final test set.

3. **Metrics**:

Most of the papers we reviewed relied primarily on accuracy ([1] through [5]). [1] noted that precision recall and F1 would have been useful as additional measures. We will use macro average F1 as a measure of overall performance.

We may also explore ratios between premise/hypothesis and hypothesis-only results in order to determine percent improvement across models and across labels.

4. **Models:**

For BERT, we will use the Hugging Face implementation. The python code in [7] and [8] will be used as a starting point.

For our LSTM model, we will use the PyTorch LSTM model from the course resources as our starting point.

We expect that we will experiment with structure of the final classifier layer. For example, [8] uses a linear layer plus a softmax layer, whereas [2] uses a stack of three tanh layers with the top layer feeding a softmax classifier.

If time permits, we will explore adding an attention mechanism to the LSTM model.

5. General Reasoning:

NLI is intended to explore whether a model, in some sense, understands the semantics/meaning of sentences. [4] argues that any interesting NLI model must therefore depend on both the premise and the hypothesis rather than relying on statistical irregularities in the hypotheses. Such irregularities can be introduced by the crowdsourcing mechanism by workers who use heuristics to create hypotheses. [4] identifies specific words, grammar and lexical semantics as types of irregularities. [5] identifies artifacts that are associated with particular category labels.

BERT which applies bidirectional training to the Transformer architecture, has achieved a greater level of success than prior models on a variety of NLP and NLU tasks. Therefore it is interesting explore how BERT performs on the NLI task relative to a hypothesis-only baseline.

6. Summary of Progress:

We have reviewed papers describing approaches to NLI ([1]-[3]) as well as papers describing challenges and limitations for these approaches ([4]-[5]).

We have reviewed approaches to contextual embedding including the course materials and the BERT paper [6].

We have reviewed a number of examples where BERT is used for classification or NLI: These include:
- The course resources notebook which uses Bert-as-a-Service and the Google implementation of BERT for classification;
- [7] which uses the Hugging Face implementation of BERT for classification;
- [8[which uses the Hugging Face implementation of BERT for NLI.

We have been able to get some rudimentary results using all three of these examples.

We are unsure, at this time, about the amount of fine tuning to use for BERT. We will start by relying on the guidance provided in the BERT paper[6], section A.3.

We are unsure about the amount of data we will be able to use given limitations on time and resources. The SNLI dataset is huge and we may need to work with only a portion in order to finish our project on time.

We have ben using Google Colab as our computing platform, but we may either upgrade to Colab Pro or switch to Azure.

References

1. Ido Dagan, Oren Glickman, Bernando Magnini, The PASCAL Recognising Textual Entailment Challenge. In Machine Learning Challenges pages 177-190.
https://u.cs.biu.ac.il/~nlp/RTE1/Proceedings/dagan_et_al.pdf

2. Samuel R. Bowman, Christopher Potts, Gabor Angeli and Christopher Manning, A large annotated corpus for learning natural language inference, 2015. In Proc of EMNLP
https://arxiv.org/abs/1508.05326

3. Tim Rocktaschel, Edward Grefenstette, Karl Moritz Herman, Tomas Kocisky and Phil Blumson, Reasoning about Entailment with Neural Attention, 2015. In ICLR 2016
https://arxiv.org/pdf/1509.06664.pdf

4. Adam Poliak, Jason Nardowsky, Aparajita Haldar, Rachel Rudinger and Benjamin Van Durme, Hypothesis Only Baselines in Natural Language Inference, 2018.In Proc of *SEM
https://www.aclweb.org/anthology/S18-2023.pdf

5. Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman and Noah A. Smith, Annotation Artifacts in Natural Language Inference Data, 2018 In Proc of NAACL.
https://arxiv.org/abs/1803.02324

6. Jacob Devlin, MIng-Wi Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Biirectional Transformers for Language Understanding, Submitted on October 11, 2018 (v1), last revised May 24, 2019
https://arxiv.org/abs/1810.04805

7. Chris Mcormick, Nick Ryan, BERT Fine-Tuning Tutorial with PyTorch, July 22, 2019, Revised 3/20/20
https://mccormickml.com/2019/07/22/BERT-fine-tuning/

8. Yang Gao, A Natural Language Inference (NLI) model based on BERT
https://github.com/yg211/bert_nli