Transferring Representations of Logical Connectives

Aaron Traylor
Dept. of Computer Science
Brown University

Ellie Pavlick
Dept. of Computer Science
Brown University

Roman Feiman
Dept. of Cognitive, Linguistic, and Psychological Sciences
Brown University

{aaron_traylor, ellie_pavlick, roman_feiman}@brown.edu

Abstract

In modern natural language processing pipelines, it is common practice to "pretrain" a generative language model on a large corpus of text, and then to "finetune" the created representations by continuing to train them on a discriminative textual inference task. However, it is not immediately clear whether the logical meaning necessary to model logical entailment is captured by language models in this paradigm. We examine this pretrain-finetune recipe with language models trained on a synthetic propositional language entailment task, and present results on test sets probing models' knowledge of axioms of first order logic.

1 Introduction

In modern natural language processing pipelines, it is common practice to "pretrain" a generative language model on a large corpus of text, and then to "finetune" the created representations by continuing to train them on a discriminative textual inference task. This pretrain-finetune recipe has led to state-of-the-art accuracy on natural language inference tasks (Wang et al., 2018), including tasks that explicitly target logical, compositional reasoning (Williams et al., 2017). However, *a priori*, it is not obvious that language model pretraining should lead models to encode anything mirroring the functions of logical concepts such as entailment, negation, and disjunction, despite their importance in language, and in fact there are many reasons why we might directly expect language modeling *not* to encode logical meaning: for example, language modeling does not provide models with access to variable bindings or truth values (a key component for defining logical functions), and it only provides models with access to positive training examples (sentences that are true in some possible world, assuming they were true at the time they were uttered) but not negative examples (sentences that have never been attested). This difference between our expectations for these models and their empirical performance gives rise to a broader question: under what conditions can core logical concepts emerge, and what information is sufficient to learn them? We assume entailment as a task is crucial to beginning to form an answer; if a system can accurately determine whether a given premise logically entails a given hypothesis, it may have a representation of logical conjunction, disjunction, conditionals, and negation. Thus, by observing how language model pretraining affects the performance of a model of logical entailment, we can better understand the extent to which these models capture logical reasoning capabilities, if at all. The primary question we aim to answer is: does the traditional language modelling objective result in representations which readily support classical logical reasoning?

Due to the pervasiveness of statistical artifacts in natural language inference datasets (Gururangan et al., 2018; Tsuchiya, 2018), directly assessing the logical reasoning capabilities of neural models using these datasets is challenging. Thus, we design a set of experiments using a toy dataset and task which uses propositional logic sentences in place of natural language, but otherwise uses the same common pretraining+finetuning recipe.

Note that we do not aim to answer the question "can neural networks do logical reasoning" in general. This question has been explored extensively elsewhere (see e.g. (Bowman et al., 2014; Geiger et al., 2018; Evans et al., 2018)). Rather, we are interested in understanding what aspects of logical reasoning can be encoded during unsupervised pretraining.

Specifically, we test whether sentence encodings learned on a language modeling task transfer to a "downstream" entailment task, i.e. improving the efficiency and/or accuracy of the model trained on the downstream task.

Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA), pages 22–25

2 Experimental Design

2.1 Dataset creation

We create two propositional logic language datasets– one for each of the pretraining and entailment steps. For the pretraining corpus, we create 500,000 unique well-formed propositional logic sentences containing atomic symbols and logical connectives, which is roughly the size of a large natural language corpus. These sentences are generated by nesting between 2 and 14 clauses, each containing a unary or binary logical connective from the set $(\neg, \models, \&, \|)$ and atomic symbols. In order to parallel a natural language pretraining corpus, we construct our data such that only logically consistent sentences are present in the propositional logic language. E.g. $A\&B$ might appear as a sentence in our corpus, whereas $A\&\neg A$ is logically inconsistent and would not be included. There are 30,000 unique symbols across our dataset, and symbols are distributed between atomic sentences uniformly, as opposed to the Zipfian distribution of natural language: this is designed in order to make it extremely challenging for the model to make judgments based on co-occurence statistics of symbols.

For the entailment training dataset, 100,000 unique propositional logic premise/hypothesis pairs are generated using the same sentence-generating algorithm as used for the pretraining dataset. As done by (Evans et al., 2018), the dataset is balanced such that, for each premise/hypothesis pair (A_1, A_2), there is a corresponding premise/hypothesis pair (B_1, B_2) such that $A_1 \models A_2$ and $B_1 \models B_2$ but $A_1 \not\models B_2$ and $B_1 \not\models B_2$. This reduces the effect of artifacts resulting from sentence generation, as each sentence is used evenly with each of the *entailed* and *not entailed* labels. We hold out an additional 5,000 pairs containing sentences not seen in training as a validation dataset.

2.2 Language model pretraining and finetuning

The objective of our model during pretraining is the typical language modelling objective. We use unidirectional LSTMs (Hochreiter and Schmidhuber, 1997), as well as Transformers (Vaswani et al., 2017) as sequence models. We train models until the perplexity converges to a local minimum, using early stopping on a validation set.

We use a simple linear classifier based on the last step of the sequence model to determine model predictions on the entailment task. We compare the performance at the entailment task of pretrained sequence models to those initialized from scratch.

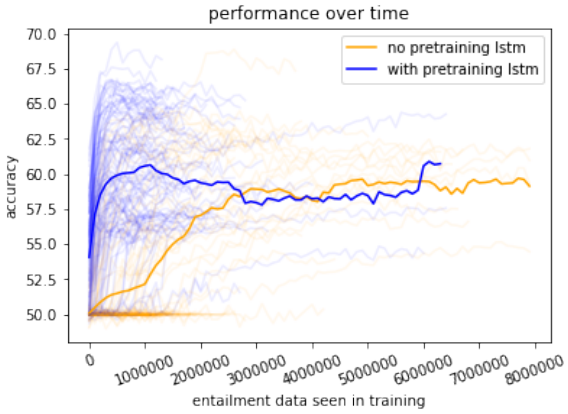

Figure 1: Average accuracy of 100 different hyperparameter settings of LSTMs on the entailment validation dataset over time.

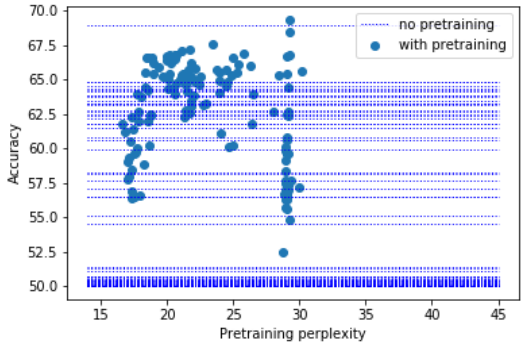

Figure 2: Perplexity versus accuracy for 100 different LSTM hyperparameter settings.

2.3 Axiomatic test sets

We create several diagnostic test datasets representing axioms of first order logic (e.g. the double negation diagnostic dataset contains entailment pairs of the structure A entails $\neg\neg A$.) These diagnostic datasets allow a fine-grained look at which functional elements of logical connectives the network is or is not able to capture after training. Sentences within a diagnostic dataset range across lengths to observe whether length significantly impacts the models' judgments. Performance on these datasets is judged by F1 score– there are several distractor negative examples based on each pattern.

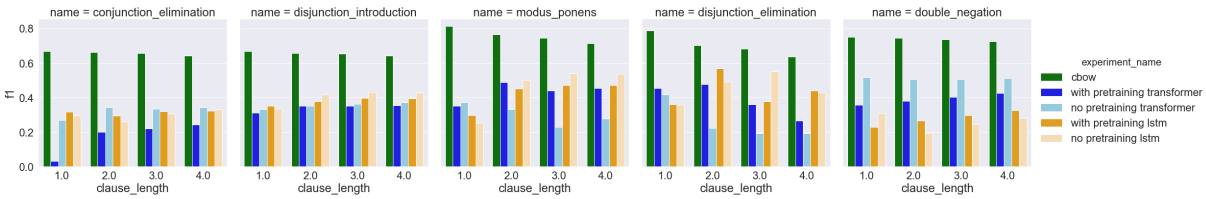

Figure 3: F1-score on axiomatic test sets of the best performing classifier model of each of the four conditions.

Model	w/ Pretraining	w/o Pretraining
CBOW	-	51.47
LSTM	70.41	68.74
Transformer	70.54	63.95

Table 1: Best performance on entailment validation set across hyperparameter sweep of each model.

3 Experiments

We compare the performance on the entailment task of a classifier on top of a model pretrained on the language modelling objective described in 2.1 to that of a classifier on top of a model with randomly initialized parameters. Models were implemented in pytorch (Paszke et al., 2019) and were trained using GPUs. We conduct a large hyperparameter sweep for each of our four settings: LSTMs and Transformers as classifier models with and without pretraining before the entailment task. We search across learning rate, hidden dimension, symbol embedding dimension, dropout, weight decay, and, respective to each model, number of stacked LSTMs and number of transformer heads/layers.

4 Results

The entailment dataset is evenly balanced, so maximum class accuracy is 50%. As shown in Table 1, the performance of the different types of classification models is comparable. The baseline continuous bag of words model performs slightly above chance.

Figure 1 displays the validation accuracy trajectory over time of 100 different hyperparameter settings for LSTM initialized with and without pretraining. Models that do not reach a new maximum accuracy within 10 epochs have their training stopped early. Most of the pretrained models' performance spikes early, while many models initialized from scratch take much longer and must see the training data many more times before their accuracy begins to increase.

Figure 2 plots the relationship between the perplexity and accuracy that LSTM models converge to. There appears to be a local minimum of perplexity that the models achieve in pretraining, and this data suggests no relationship between lower perplexity of the language model and higher accuracy.

The models do not perform well on the diagnostic axiomatic test set dataset across the board, as observed in Figure 3. Despite the extensive training in each scenario, the models are unable to regularly capture even extremely simple patterns, such as double negation of one clause. The effect of clause length varies– the models may be picking up on dataset artifacts for shorter premises and hypotheses. Furthermore, despite its low performance at the entailment task, the F1 score of the continuous bag of words model is significantly higher than that of the other models. This is an interesting trend that could be due to the pretraining leading the model into a poor initialization, which requires further investigation.

5 Discussion

In our experiments, we find mixed evidence to suggest that language model pretraining on sentences gives a reliable performance increase over models trained entirely from scratch with regards to purely logical entailment. On average, pretrained LSTMs and Transformers perform slightly better than those initialized entirely from scratch, but the performance of these models at held-out test sets which specifically probe fundamental axioms of logical entailment is underwhelming and worse than the baseline model. Furthermore, no model performs especially well at the entailment task– logical operators are applied inflexibly to their input, and no model performs near 100% accuracy.

It is potentially the case that, within modern natural language inference pipelines, the logical information encoded at the pretraining step is partially responsible for the increase in accuracy at downstream tasks. However, it is hard to say that any such transfer of representations that facilitate

explicit logical capabilities occurs. It is possible that language models, if they do not preserve logical information, lead models to perform well at entailment tasks in general by capturing complex lexical heuristics and associations between topics at the pretraining step.

In future work, we will test whether pretraining on both positive and negative examples affects entailment performance. Furthermore, we will test whether adjusting from a uniform distribution of the symbols to a Zipfian distribution causes the performance of the pretrained model to improve, which would suggest that these models rely much more heavily on exploiting frequency heuristics than captured logical capabilities to more accurately model the training data.

References

Samuel R Bowman, Christopher Potts, and Christopher D Manning. 2014. Recursive neural networks can learn logical semantics. *arXiv preprint arXiv:1406.1827*.

Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. 2018. Can neural networks understand logical entailment? *arXiv preprint arXiv:1802.08535*.

Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2018. Stress-testing neural models of natural language inference with multiply-quantified sentences. *arXiv preprint arXiv:1810.13033*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. *arXiv preprint arXiv:1804.08117*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.