Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by generative pre-training of a language model on a diverse corpus of unlabeled text, followed by discriminative fine-tuning on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on question answering (RACE), and 1.5% on textual entailment (MultiNLI). 1

Introduction

The ability to learn effectively from raw text is crucial to alleviating the dependence on supervised learning in natural language processing. Most deep learning methods require substantial amounts of manually labeled data. In these situations, models that can leverage linguistic information from unlabeled data provide a valuable alternative to gathering more annotation, which can be time-consum.Leveraging more than word-level information from unlabeled text is challenging for two main reasons. First, it is unclear what type of information is needed. Second, learning good representations in an unsupervised fashion can provide a significant performance boost. The most compelling evidence for this so far has been the extensive use of pre- trained word embeddings.There is no consensus on the most effective way to transfer these learned representations to the target task. Existing techniques involve a combination of making task-specific changes to the mode and learning text representations that are useful for transfer. Recent research has looked at various objectives such as language modeling, machine translation, and discourse coherence.We explore a semi-supervised approach for language understanding tasks. We use a combination of unsupervised pre-training and supervised fine-tuning. Our goal is to learn a universal representation of the language language. Preprint. l architecture [43, 44, 45]We assume access to a large corpus of unlabeled text and several datasets with manually annotated training examples (target tasks) We employ a two-stage training procedure. First, we use a language modeling objective on the data to learn the initial parameters of a neural network model. Subsequently, we adapt these parameters to a target that transfers with little adaptation to a wide range of tasks.We use the Transformer model architecture, which has been shown to perform strongly on various tasks such as machine translation. This model choice provides us with a more structured memory for handling long-term dependencies in text, compared to alternatives like recurrent networks. During transfer, we utilize task-specifiiction. task using the corresponding supervised objective.We evaluate our approach on four types of language understanding tasks. These include natural language inference, question answering, semantic similarity, and text classification. Our general task-agnostic model outperform all other models on all four tasks. We demonstrate that we can fine-tune effectively with minimal changes to the architecture of the model.We train discriminatively trained models that employ architectures specifically crafted for each task. We significantly improve upon the state of the art in 9 out of the 12 tasks studied. We achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test) and 5.5% on the GLUE multi-task benchmark.

Related Work

Our work broadly falls under the category of semi-supervised learning for natural language. This paradigm has attracted significant interest, with applications to tasks like sequence labeling or text classification. The earliest approaches used unlabeled data to compute word-level or phrase-level statistics, which were then used as features in a supervised model. Over the last few years, researchers have demonstrated the benefits of using word level statistics.Recent approaches have investigated learning and utilizing more than word-level semantics from unlabeled data. Phrase-level or sentence-level embeddings have been used to encode text into suitable vector representations. These approaches, however, mainly transfer word- level information, whereas we aim to capture higher- level semantics.Unsupervised pre-training is a special case of semi-supervised learning. The goal is to find a good initialization point instead of modifying the supervised learning objective. Early works explored the use of the technique in image classification and regression tasks. Research demonstrated that pre- training acts as a regularization scheme, enabling better generalization in deep neural netwThe method has been used to help train deep neural networks on various tasks like image classification and speech recognition. The closest line of work to ours involves pre-training a neural network using a language modeling objective and then fine-tuning it on a target task with supervision. Dai et al. [13] and Howard and Ruder [21] follow this method to improve text classification.Transformer networks allow us to capture longer- range linguistic structure. We also demonstrate the effectiveness of our model on a wider range of tasks including natural language inference, paraphrase detection and story completion. Other approaches [43, 44, 38] use hidden representations from a 2 pre-trained l.Add auxiliary unsupervised training objectives is an alternative form of semi-supervised learning. This involves a substantial amount of new parameters for each separate target task, whereas we require minimal changes to our model architecture during transfer. Early work by Collobert and Weston [10] used a wide variety of auxiliary NLP tasks.

Framework

Unsupervised pre-training

Given an unsupervised corpus of tokens U = {u1, . . . , un}, we use a standard language modeling objective to maximize the following likelihood. The conditional probability P is modeled using a neural network with parameters Th. These parameters are trained using stochastic gradient descent [51].

Supervised fine-tuning

We perform experiments on a variety of supervised tasks including natural language inference, question answering, semantic similarity, and text classification. Some of these tasks are available as part of the recently released GLUE multi-task benchmark, which we make use of. Figure 1 provides an overview of all the tasks and datasets.The task remains challenging due to the presence of a wide variety of phenomena like lexical entailment, coreference, and lexical and syntactic ambiguity. We evaluate on five datasets with diverse sources, including image captions (SNLI), transcribed speech, popular fiction, and government reports.Our method significantly outperforms the baselines on four of the five datasets. We achieve absolute improvements of upto 1.5% on MNLI, 5% on SciTail, 5.8% on QNLI and 0.6% on SNLI. This demonstrates our model's ability to better reason over multiple sentences, and handle aspects of linguistic ambiguity.On (2490 examples), we achieve an accuracy of 56%, which is below the 61.7% reported by a multi-task biLSTM model. Given the strong performance of our approach on larger NLI datasets, it is likely our model will benefit from multi- task training as well.Results done using the GLUE benchmark. (mc= Mathews correlation, acc=Accuracy, pc=Pearson correlation) Method Classification Semantic Similarity GLUE CoLA SST2 MRPC STSB QQP (mc) (acc) (F1) (pc) (F1)

Task-specific input transformations

For some tasks, like text classification, we can directly fine-tune our model as described above. Certain other tasks have structured inputs such as ordered sentence pairs, or triplets of document, question, and answers. Since our pre-trained model was trained on contiguous sequences of text, we require some modifications to apply it to these tasks.We use a traversal-style approach where we convert structured inputs into an ordered sequence that our pre-trained model can process. These input transformations allow us to avoid making extensive changes to the architecture across tasks. We provide a brief description of these input transformations below and Figure 1 provides a visual i.For entailment tasks, we concatenate the premise p and hypothesis h token sequences, with a delimiter token ($) in between. For similarity tasks, there is no inherent ordering of the two sentences being compared. To reflect this, we modify the input sequence to contain both possible sentence orderings.We are given a context document z, a question q, and a set of possible answers {ak}. We concatenate the document context and question with each possible answer. Each of these sequences are processed independently with our model and then normalized via a softmax layer to produce an out.

Experiments

Setup

We use the BooksCorpus dataset [71] for training the language model. It contains over 7,000 unique unpublished books from a variety of genres including Adventure, Fantasy, and Romance. Crucially, it contains long stretches of contiguous text, which allows the generative model to learn to condition on long-range information.Our language model achieves a very low token level perplexity of 18.4 on this corpus. It is shuffled at a sentence level - destroying long-range structure. Mo. t tasks and datasets used in our experiments were: Natural language inference SNLI, MultiNLI, Question NLI, RTE, SciTail.We trained a 12-layer decoder-only transformer with masked self-attention heads. For the position-wise feed-forward networks, we used 3072 dimensional inner states. We used the Adam optimization scheme [27] with a max learning rate of 2.5e-4.We used a bytepair encoding (BPE) vocabulary with 40,000 merges and residual, embedding, and attention dropouts with a rate of 0.1 for regularization. We also employed a modified version of L2 regularization proposed in [37], with w = 0.01 on all non bias or gain weights. e used the Gaussian Error Linear Unit (GELU) We used learned position embeddings instead of the sinusoidal version proposed in the original work. We use the ftfy library2 to clean the raw text in BooksCorpus, standardize some punctuation and whitespace, and use the spaCy tokenizer.

Analysis

Figure 2 illustrates the performance of our approach on MultiNLI and RACE as a function of the number of layers transferred. We observe the standard result that transferring embeddings improves performance and that each transformer layer provides further benefits up to 9% for full transfer. This indicates that each layer inFigure 2: Effect of transferring increasing number of layers from the pre-trained language model on RACE and MultiNLI. Performance per task is normalized between a random guess baseline and the current state-of-the-art with a single model. We'd like to better understand why language mOdel pre-training of transform- ers is effective. A hypothesis is that the underlying generative model learns to perform many of the tasks we evaluate on in order to improve its language modeling capability and that the more structured 7 7 Table 5: Analysis of various model ablations on different tasks.We designed a series of heuristic solutions that use the underlying generative model to perform tasks without supervised finetuning. We visualize the transformer and LSTM to help people understand the difference between the two. We also show that the transformer assists in transfer compared to LSTMs.The effectiveness of these heuristic solutions over the course of generative pre-training in Fig 2(right) is stable and steadily increases over training. We observe the LSTM exhibits higher variance in its zero-shot performance suggesting that the inductive bias of the Transformer architecture assists in transfer. For CoLA (linguistic acceptability), examples are scored as the average token log-probability the generative model assigns and predictions are made by thresholding. For SST-2 (sentiment analysis), we append the token very to each example and restrict the language model's output distribution to only the words positive and negative and guess the token it assigns higher probability to as the prediction.We perform three different ablation studies (Table 5) First, we examine the performance of our method without the auxiliary LM objective during fine-tuning. We observe that the auxiliary objective helps on the NLI tasks a bit more than the LM objective. Second, we predict the resolution that the generative model assigns higher average token log-probability to the rest of the sequence.The trend suggests that larger datasets benefit from the auxiliary objective but smaller datasets do not. We observe a 5.6 average score drop when using the LSTM instead of the Transformer. We also compare with our transformer architecture directly trained on supervised target tasks, without pr. nd QQP.

Conclusion

We introduce a framework for achieving strong natural language understanding with a single task-agnostic model. By pre-training on a diverse corpus with long stretches of contiguous text our model acquires significant world knowledge and ability to process long-range dependencies. This knowledge is then successfully transferred to solving discriminative tasks.Using unsupervised (pre-)training to boost performance on discriminative tasks has long been an important goal of Machine Learning research. Our work suggests that achieving significant performance gains is indeed possible, and offers hints as to what models (Transformers) and data sets (text with long range dependencies) work best.

Figures

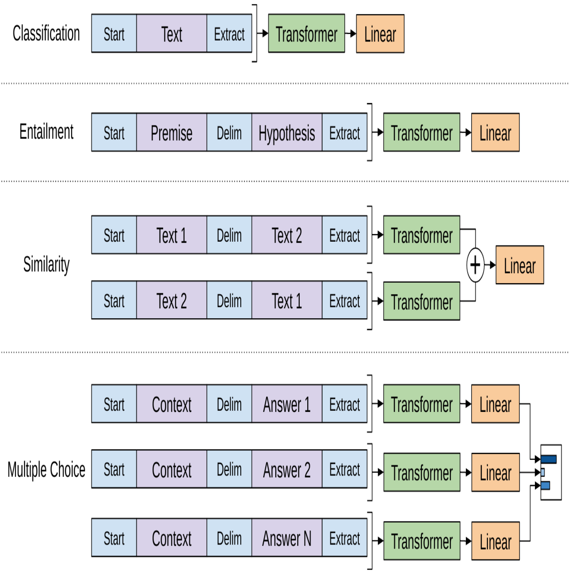

Figure 1

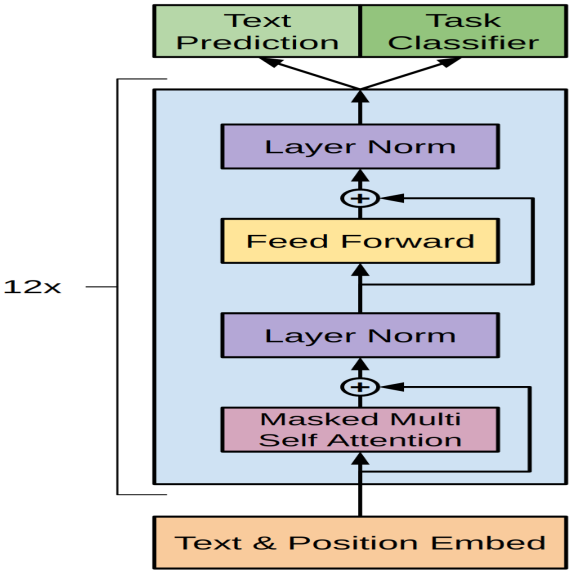

Figure 2

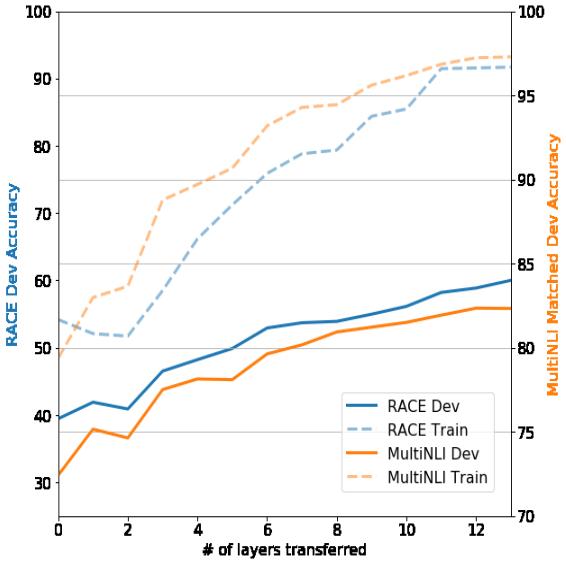

Figure 3

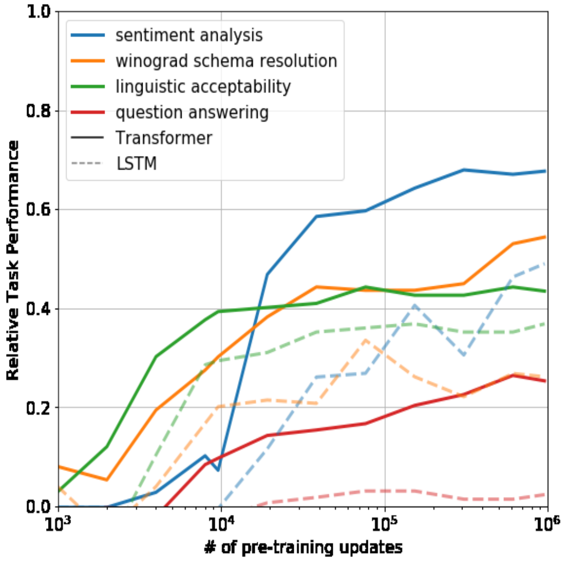

Figure 4