SPAN SELECTION PRE-TRAINING FOR QUESTION ANSWERING

A PREPRINT

Michael Glass,[1] **Alfio Gliozzo,**[1] **Rishav Chakravarti,**[1] **Anthony Ferritto,**[1]
Lin Pan,[1] **G P Shrivatsa Bhargav,**[2] **Dinesh Garg,**[1] **Avirup Sil**[1]
[1] IBM Research AI
[2] Dept. of CSA, IISC, Bangalore
mrglass@us.ibm.com, gliozzo@us.ibm.com, rchakravarti@us.ibm.com, aferritto@ibm.com,
panl@us.ibm.com, bhargavs@iisc.ac.in, garg.dinesh@in.ibm.com, avi@us.ibm.com

September 11, 2019

ABSTRACT

BERT (Bidirectional Encoder Representations from Transformers) and related pre-trained Transformers have provided large gains across many language understanding tasks, achieving a new state-of-the-art (SOTA). BERT is pre-trained on two auxiliary tasks: Masked Language Model and Next Sentence Prediction. In this paper we introduce a new pre-training task inspired by reading comprehension and an effort to avoid encoding general knowledge in the transformer network itself. We find significant and consistent improvements over both $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ on multiple reading comprehension (MRC) and paraphrasing datasets. Specifically, our proposed model has strong empirical evidence as it obtains SOTA results on Natural Questions, a new benchmark MRC dataset, outperforming $\text{BERT}_{\text{LARGE}}$ by 3 F1 points on short answer prediction. We also establish a new SOTA in HotpotQA, improving answer prediction F1 by 4 F1 points and supporting fact prediction by 1 F1 point. Moreover, we show that our pre-training approach is particularly effective when training data is limited, improving the learning curve by a large amount.

1 Introduction

State-of-the-art approaches for NLP tasks are based on language models that are pre-trained on tasks which do not require labeled data [Peters et al., 2018, Howard and Ruder, 2018, Devlin et al., 2018, Yang et al., 2019, Liu et al., 2019, Sun et al., 2019]. Fine tuning language models to downstream tasks, such as question answering or sentence paraphrasing, has been shown to be a general and effective strategy. BERT is a recently introduced and highly successful model for language understanding.

The general BERT adaptation approach is to alter the model used for pre-training while retaining the transformer encoder layers. The model discards the layers used for the final prediction in the pre-training tasks and adds layers to predict the target task. All parameters are then fine tuned on the target task.

BERT is based on the transformer architecture [Vaswani et al., 2017], and trained on the following two unsupervised tasks:

- Masked Language Model (MLM): predicting masked word pieces from the surrounding context
- Next Sentence Prediction (NSP): predicting if the two provided sequences follow sequentially in text or not

然而，完形填空的预训练任务通常只需要浅显的预测，或者需要记忆的知识。

对于许多完形填空实例，模型只需要对语法或词法的理解来回答。

The masked LM or "cloze" task [Taylor, 1953] and next sentence prediction are auxiliary tasks [Ando and Zhang, 2005] requiring language understanding, and therefore train the model to acquire effective representations of language. However, the cloze pre-training task often poses instances that require only shallow prediction, or else require memorized knowledge. For many cloze instances the model simply requires syntactic or lexical understanding to answer. For example, in the cloze instances in Table 1 the first two rows require syntactic and lexical understanding respectively. Other cloze instances mainly require completing collocations, as in the third example. However, some cloze instances require general knowledge, as in the last instance, which essentially asks where Hadrian died.

Type	Cloze
Syntactic	In the 15th century, the blast furnace spread into what is now Belgium where it was improved.
Lexical	Akebia quinata grows to 10 m (30 ft) or more in height and has compound leaves with five leaflets.
Collocation	Apollo 11 was launched by a Saturn V rocket from Kennedy Space Center on Merritt Island, Florida
Memorized Knowledge	Hadrian died the same year at Baiae, and Antoninus had him deified, despite opposition from the Senate.

Table 1: Cloze instances of different types

Other language models face the same challenge. In GPT-2 [Radford et al., 2019] the entities present in a language generation prompt are expanded with related entities. For example, in a prompt about nuclear materials being stolen on a Cincinnati train, GPT-2 references "Ohio news outlets", "U.S. Department of Energy", and "Federal Railroad Administration" in ways consistent with their real world relationships to the entities in the prompt.

预先训练过的语言模型具有回答问题和生成文本的能力，文本中包含实体，这些实体以与现实世界的关系相对应的方式连接在文本中。我们将这种能力称为“一般知识”。

Pre-trained language models have some capability to answer questions and to generate text containing entities connected in the text in ways that correspond to their real world relationships. We term this capability "general knowledge".

以前的解决办法是将knowledge直接表征为稠密的激活参数矩阵，这样需要大量的数据时间来积累这些general knowledge

Unfortunately, in these state-of-the-art transformer architectures this knowledge is represented in the densely activated parameter matrices. Consequently, the model must be very large to accommodate a substantial amount of general knowledge, leading to long pre-training times even with high end hardware. However, for any given instance in a task, only a tiny fraction of this knowledge is relevant. One possible solution is to offload the requirement of general knowledge to a sparsely activated network, which can then bring in relevant knowledge on a per-instance basis. Previous work also found seemingly limitless improvements from increasing model capacity [Shazeer et al., 2017], which is possible only through sparse activation.

As a step towards this goal, we use *span selection* as an additional auxiliary task. This task is similar to the cloze task, but is designed to have fewer simple instances requiring only syntactic or collocation understanding. For cloze instances that require general knowledge, rather than forcing the model to encode this knowledge in its parameterization, we provide a relevant and answer-bearing passage to play the role of an instance specific information retrieval system.

cloze实例要求general knowledge,不是将general knowledge强制编码到参数中，而是提供一个相关的并且答案导向的，起到特定信息检索系统的实例角色的passage

We provide an extensive evaluation of the span selection pre-training method across six tasks: the first four are reading comprehension based: the Stanford Question Answering Dataset (SQuAD) in both version 1.1 and 2.0; followed by the Google Natural Questions dataset [Kwiatkowski et al., 2019] and a multi-hop Question Answering dataset, HotpotQA [Yang et al., 2018]. The other two are paraphrase datasets: the Microsoft Research Paraphrasing Corpus (MRPC) and the Quora Question Paraphrasing (QQP) dataset. We report consistent improvements over both $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ models in the reading comprehension benchmarks and some positive signal in paraphrasing.

The rest of the paper is structured as follows. We describe earlier work on similar tasks and relate our extended pre-training to the broader research efforts on pre-training transformers. To provide context for our contribution, we review the most relevant parts of BERT. Next, we describe and formalize our pre-training task and the architectural adjustments to BERT. Finally we provide an extensive empirical evaluation in downstream tasks. We consider six tasks in total, four QA tasks and two paraphrasing tasks.

2 Related Work

We review related work in three categories: other efforts to use automatically constructed tasks similar to extractive QA, research towards adding new pre-training tasks, and work that extends the pre-training with more data.

Previous work has explored tasks similar to span selection pre-training. These are typically cast as approaches to augment the training data for question answering systems, rather than alleviating the need for encoding world knowledge in a language model or general pre-training for language understanding.

Hermann et al. [2015] introduces a reading comprehension task constructed automatically from news articles with summaries. In this view the constructed dataset is used both for training and test. Also, entities were replaced with anonymized markers to limit the influence of world knowledge. Unlike our span selection pre-training task, this requires summaries paired with articles and focuses only on entities. A similar approach was taken by Dhingra et al. [2018] to augment training data for question answering. Wikipedia articles were divided into introduction and body with sentences from the introduction used to construct queries for the body passage. Phrases and entities are used as possible answer terms.

Onishi et al. [2016] constructed a question answering dataset where answers are always people. Unlike other work, this did not use document structure but instead used a search index to retrieve a related passage for a given question. Because the answers are always people, and there are only a few different people in each passage, the task is multiple choice rather than span selection. Self training [Sachan and Xing, 2018] has also been used to jointly train to construct questions and generate self-supervised training data.

Since the development of BERT there have been many efforts towards adding or modifying the pre-training tasks. Joshi et al. [2019] introduced SpanBERT, a task that predicts the tokens in a span from the boundary token representations. Note that, unlike span selection, there is no relevant passage used to select an answer span. ERNIE 2.0 [Sun et al., 2019] trained a transformer language model with seven different pre-training tasks, including a variant of masked language model and a generalization of next-sentence prediction. XLNet [Yang et al., 2019] introduced the permuted language model task, although it is not clear whether the success of the model is due to the innovative pre-training or larger quantity of pre-training.

BERT was trained for one million batches, with 256 token sequences in each. Although this is already a considerable amount of pre-training, recent research has shown continued improvement from additional pre-training data. XLNet [Yang et al., 2019] used four times as much text, augmenting the Wikipedia and BooksCorpus [Zhu et al., 2015] with text from web crawls, the number of instances trained over was also increased by a factor of four. RoBERTa [Liu et al., 2019] enlarged the text corpus by a factor of ten and trained over fifteen times as many instances. This, along with careful tuning of the MLM task resulted in substantial gains. Unfortunately, these very large-scale pre-training approaches require significant hardware resources. We restrict our experiments to extended pre-training with less than half the steps of BERT (390k batches of 256).

3 Background

In this section, we give the readers a brief overview of the BERT [Devlin et al., 2018] pre-training strategy and some details which we modify for our novel span selection auxiliary task.

3.1 Architecture and setup

BERT uses a transformer [Vaswani et al., 2017] architecture with L layers and each block uses A self-attention heads with hidden dimension H. The input to BERT is a concatenation of two segments $x_1, \ldots, x_M$ and $y_1, \ldots, y_N$ separated by special delimiter markers like so: $[CLS], x_1, \ldots, x_M, [SEP], y_1, \ldots, y_N, [SEP]$ such that $M + N < S$ where S is the maximum sequence length allowed during training[1]. This is first pre-trained on a large amount of unlabeled data and then fine-tuned on downstream tasks which has labeled data.

3.2 Objective functions

BERT used two objective functions during pre-training: masked language modeling and next sentence prediction. We discuss them in brief.

Masked Language Model (MLM): A random sample of the tokens in the input sequence is replaced with a special token called $[MASK]$. MLM computes a cross-entropy loss on predicting these masked tokens. Particularly, BERT selects 15% of the input tokens uniformly to be replaced. 80% of these selected tokens are replaced with [MASK] while 10% are left unchanged, and 10% are replaced with random token from the vocabulary.

Next Sentence Prediction (NSP): This is a binary classification loss that predicts if two sentences follow each other in the original text. The examples are sampled with equal probability such that positive examples are consecutive sentences while negatives are artificially created by adding sentences from different documents.

[1] We follow standard notation here as in previous work.

4 Our Proposed Framework

In the previous section we briefly discussed the BERT framework along with its objective functions. In this section, we will propose a novel pre-training task for bi-directional language models called span selection.

4.1 Span Selection

Span selection is a pre-training task inspired both by the reading comprehension task and the limitations of cloze pre-training. Figure 1 illustrates an example of a span selection instance. The *query* is a sentence drawn from a corpus with a term replaced with a special token: [BLANK]. The term replaced by the blank is the *answer term*. The *passage* is relevant as determined by a BM25 [Robertson and Zaragoza, 2009] search, and answer-bearing (containing the answer term).

Query	"In a station of the metro" is an Imagist poem by [BLANK] published in 1913 in the literary magazine Poetry
Passage	... Ezra Pound 's famous Imagist poem, "In a station of the metro", was inspired by this station ...
Answer Term	Ezra Pound

Figure 1: Example Span Selection Instance

Unlike BERT's cloze task, where the answer must be drawn from the model itself, the answer is found in a passage using language understanding.

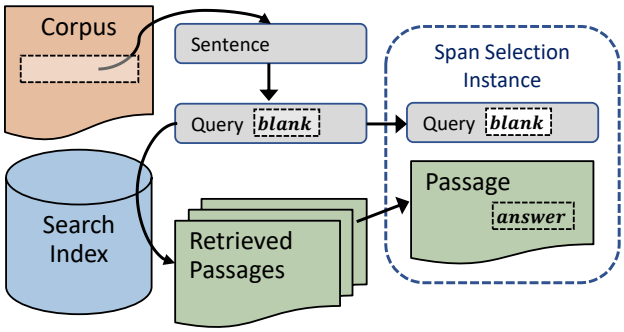

Figure 2: Span Selection Training Generation

Figure 2 outlines the process of generating span selection pre-training data. The input is an unlabeled corpus, which is then split into passages and indexed. We used passages from Wikipedia[2] 300 to 2000 characters long, split on paragraph boundaries, and Lucene[3] 7.4.0 as the search engine. In addition to the text of the passage, we store the document ID, so that we may filter passages that occur in the same document as the query.

To gather queries, we iterate over the sentences in the corpus between 50 and 250 characters long. For each sentence, we choose an answer term to replace with a blank. We used a set of simple heuristic criteria to identify answer terms that are likely to result in queries that require deep understanding to answer: the term should be between 4 and 30 characters and either a single token from an open class part-of-speech (20%) or a noun phrase or entity (80%), as detected by a part-of-speech pattern and ClearNLP NER.

To identify the passages, we use the generated query, with the answer term removed, as a bag-of-words query to search into the passage index. The top ten results were searched for an answer-bearing passage; if none were found the query was either discarded or sampled to maintain a 30% composition of *impossible* span selection instances. The impossible instances are those that do not have the answer-term in the provided passage. We further required a minimum BM25 score of 25 (tuned manually to reflect high relevance) and a *maximum* BM25 score of 75 to avoid near duplicate text. If the answer term was part of a longer sequence of tokens shared by the query and passage, we extended the answer term to be the longest such sequence. This avoids cases where the answer term can be found through trivial surface-level matching.

[2]December 2018 snapshot

[3]`http://lucene.apache.org/`

Type	Span Selection Instance
Phrase Multiple Choice	**Q:** The year 1994 was proclaimed [BLANK] of the Family by the United Nations General Assembly. **P:** The International Year for the Culture of Peace was designated by the United Nations as the year 2000, with the aim of celebrating and encouraging a culture of peace. ...
Suggestive Inference	**Q:** On the island of Kaja in [BLANK], a male orangutan was observed using a pole apparently trying to spear or bludgeon fish. **P:** ... Although similar swamps can be found in Borneo, wild Bornean orangutans have not been seen using these types of tools.
Justified Inference	**Q:** The company's headquarters are located in the city of Redlands, California, 50 miles east of [BLANK]. **P:** Redlands (Serrano: Tukut) is a city in San Bernardino County, California, United States. It is a part of the Greater Los Angeles area. ...

Table 2: Span Selection instances of different types

Table 2 shows examples of span selection instances of different types. Rather than discreet types, these are best understood as a continuum. Comparing to the cloze types in Table 1, we see an analogy between the lexical cloze type and phrase multiple choice. These two types involve understanding what words (or phrases) are reasonable in the context from the set of wordpieces (or possible spans). The memorized knowledge cloze type contrasts with the suggestive or justified inference span selection types. Because a suggestive or justifying passage is present, the model only needs to understand language, rather than memorize facts. Simple syntactic instances are largely eliminated because closed class words are not possible answer terms. Also, since answer terms are expanded to the longest shared subsequence between query and passage, collocation instances are not a concern.

4.2 Extended Pre-training

Rather than training a transformer architecture from scratch, we initialize from the pre-trained BERT models [Devlin et al., 2018] and extend the pre-training with the span selection auxiliary task. We refer to the resulting models as $\text{BERT}_{\text{BASE}}$+SSPT (Span Selection Pre-Training) and $\text{BERT}_{\text{LARGE}}$+SSPT. We used batch sizes of 256, and a learn rate of 5e-5. All models were trained over 100 million span selection instances. We found continued improvement from 50 million to 100 million and have not yet tried larger pre-training runs. Unlike the efforts of XLNet or RoBERTa which increased training by a factor of ten relative to BERT, the additional data in SSPT represents less than a 40% increase in the pre-training of the transformer. This pre-training is also done over Wikipedia, adding no new text to the pre-training.

Figure 3 illustrates the adaptation of BERT to SSPT. The query and passage are concatenated in the standard two sequence representation, with a preceding [CLS] token and a separating [SEP] token, producing a sequence of tokens T. BERT produces output vectors for these tokens to obtain a sequence $\{\boldsymbol{v}_i\}_{i=1}^{|T|}$ of d dimensional vectors.

In span selection extended pre-training, we alter the vocabulary of the tokenizer, introducing the new special token: '[BLANK]'. We use the BertForQuestionAnswering[4] model, which uses a pointer network to find the answer location. The pointer network applies a simple fully connected network to predict the probability of start and end span pointers at each token position, using the output of the final transformer layer at that position. The loss in training is the cross entropy of these predictions with the true positions of the start and end.

Formally, The start of the answer span is predicted as $p(i = \langle start \rangle) = sigmoid(\boldsymbol{w}_{\langle start \rangle}^{\top} \boldsymbol{v}_i + b_{\langle start \rangle})$, where $\boldsymbol{w}_{\langle start \rangle} \in \mathbb{R}^d$, $b_{\langle start \rangle} \in \mathbb{R}$ are trainable parameters. Then end of the span is predicted the same way: $p(i = \langle end \rangle) = sigmoid(\boldsymbol{w}_{\langle end \rangle}^{\top} \boldsymbol{v}_i + b_{\langle end \rangle})$.

Span selection pre-training may optionally include a classifier for answerability. This is useful in pre-training sequence-pair classification tasks like paraphrasing. If the answerability classifier is included in the pre-training then the presence of the answer span in the passage is predicted with probability given by: $p(possible) = sigmoid(\boldsymbol{w}_{CLS}^{\top} \boldsymbol{v}_{CLS} + b_{CLS})$. If it is not included, for impossible instances the target prediction is for both start and end to be position zero, the

[4] `https://github.com/huggingface/pytorch-transformers`

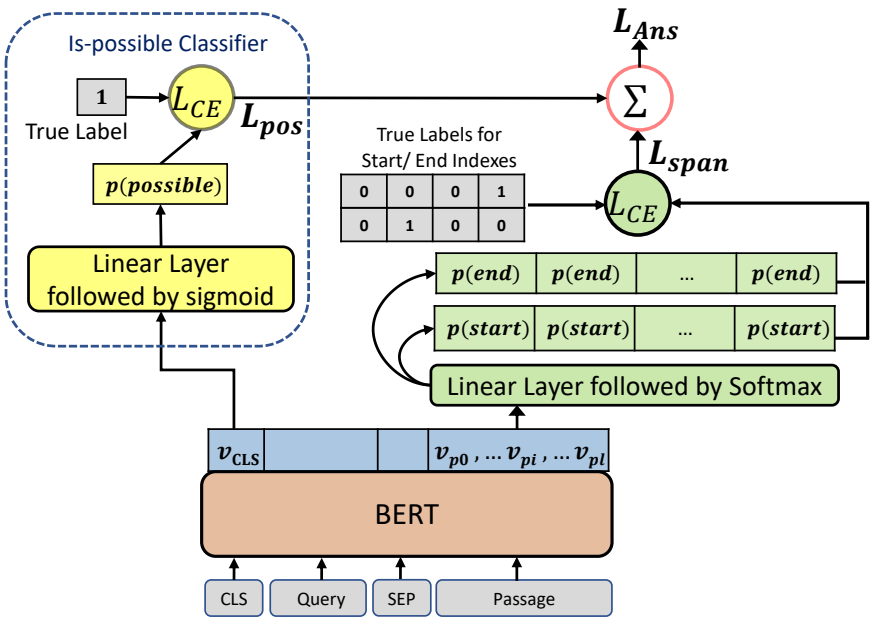

Figure 3: BERT for QA with is-possible prediction

[CLS] token. We train models for QA without the answerability classifier for 100 million instances, and models for paraphrasing with the classifier for 10 million instances.

Code to generate the data and extend pre-training is available as open source[5].

5 Downstream Tasks

Dataset	Context	Answer Types	Question Creation	Training Size	Dev Size	Test Size	Gap to Human Performance†
SQuAD 1.1	passage	span	generated	88k	11k	10k	$< 0\%$
SQuAD 2.0	passage	span, impossible	generated	130k	12k	9k	$< 0\%$
Natural Questions	document	span,yes,no, impossible	natural	307k	8k	8k	16%
HotpotQA	passage set	span,yes,no	generated	91k	7k	7k	11%

Table 3: Comparison of QA Datasets. †As of Sept. 2019

We follow previous work and evaluate our SSPT architecture on several downstream tasks. Our primary motivation is to improve question answering by improving the pre-trained language model. We also find benefit for another language understanding task: paraphrasing. Our QA benchmarks are the following:

1. Stanford Question Answering Dataset (SQuAD) v1.1 [Rajpurkar et al., 2016] and v2.0 [Rajpurkar et al., 2018]
2. Natural Questions (NQ) [Kwiatkowski et al., 2019]
3. HotpotQA [Yang et al., 2018]

The three datasets provide different characteristics of question answering and machine reading comprehension tasks as well as an opportunity to compare results with active leader-boards. Table 3 provides a summary comparison. We briefly discuss them here:

[5] `https://github.com/IBM/span-selection-pretraining`

5.1 SQuAD

SQuAD provides a paragraph of context and asks several questions about it. The task is extractive QA where the system must find the span of the correct answer from the context. We evaluate on two versions of SQuAD: v1.1 and v2.0. In v1.1 the context always contains an answer. However, in v2.0 the task contains additional questions to which the given context does not have the correct answer.

Just as in Figure 3, the question and passage are concatenated with the separators ([CLS] and [SEP]) to form the input to the pre-trained BERT. The final token representations are then used to predict the probability for each token that it is the start or end of the answer span. The span with the highest predicted probability is then the predicted answer.

5.2 Natural Questions

NQ is a dataset of over 300,000 queries sampled from live users on the Google search engine for which a Wikipedia article is contained in the top ranking search results. Crowd sourced annotators are then tasked with highlighting a short answer span to each question[6], if available, from the Wikipedia article as well as a long answer span (which is generally the most immediate HTML paragraph, list, or table span containing the short answer span), if available.

Similar to SQuAD 2.0 the NQ dataset forces models to make an attempt at "knowing what they don't know" in order to detect and avoid providing answers to unanswerable questions. In addition, the fact that the questions were encountered naturally from actual users removes some of the observational bias that appears in the artificially created SQuAD questions. Both these aspects along with the recency of the task's publication means that this is still a challenging task with lots of headroom between human performance and the best performing automated system.

Experiments on the NQ dataset use the strategies and model described by Alberti et al. [2019b] to fine tune a $\text{BERT}_{\text{LARGE}}$ model with a final layer for answerability prediction as well as sequence start/end prediction. Similar to their best performing systems, the model is first trained using the SQuAD v1.1 data set and then subsequently trained on the NQ task[7]. The hyperparameters follow Alberti et al. [2019b] with the exception of learning rate and batch size which are chosen according to the approach outlined by Smith [2018] using a 20% sub-sample of the data for each experimental setting.

5.3 HotpotQA

Recently, Yang et al. [2018] released a new dataset, called HotpotQA, for the task of reading comprehension style extractive QA. Each training instance in the *distractor* setting of this dataset comprises a question, a set of ten passages, an answer, and a binary label for each sentence in the passage-set stating whether that sentence serves as a supporting fact (or not) to arrive at the correct answer. The task is to predict both the correct answer as well as the supporting facts for any given test instance. The signature characteristic of this dataset lies in the fact that each question requires a minimum of two supporting facts from two different passages in order to derive its correct answer. Thus, this dataset tests the cross-passage, multi-hop reasoning capability of a reading comprehension based question answering system.

Our system for HotpotQA uses a three-phase approach. First, representations of the individual passages are built with a pre-trained transformer encoder. Second, interactions between these passages are attended to using a relatively shallow *global* transformer encoder. The supporting facts are predicted from the sentence representations produced by this global layer. Finally, the predicted supporting facts are then merged into a *pseudo-passage* that is used by a slightly altered version of the model for SQuAD. The one addition is that this model also predicts an answer-type ($\{yes, no, span\}$) from the [CLS] token vector.

5.4 Paraphrasing Tasks

In order to explore the wider applicability of span selection pre-training, we conducted experiments on two paraphrasing datasets, MRPC (Microsoft Research Paraphrasing Corpus) and QQP (Quora Question Pairs) as distributed in the GLUE [Wang et al., 2018] benchmark. MRPC and QQP test semantic equivalence of statements and questions respectively.

[6] Around 1% of the questions are answered as a simple Yes or No rather than a span of short answer text. Due to their small proportion, the models in this paper do not produce Yes/No answers

[7] Skipping the SQuAD v1.1 fine-tuning step for the NQ task leads to the same conclusions with respect to SSPT pre-training, but decreases the overall performance for both $\text{BERT}_{\text{LARGE}}$ and $\text{BERT}_{\text{LARGE}}$+SSPT

6 Experiments

Tables 4, 5, 6, and 7 show our results on the development set with extended span selection pre-training for BERT relative to the pre-trained BERT. We use the same hyperparameters on these tasks as the original BERT. The best results for each dataset are in bold when significant relative to the BERT baseline. The four question answering datasets are improved substantially with span selection pre-training. MRPC is also improved by span selection, while QQP performance is flat.

Method	**SQUAD 1.1**		**SQUAD 2.0**	
	F1	Exact	F1	Exact
$\text{BERT}_{\text{BASE}}$	88.524	81.220	76.453	73.292
+SSPT	**91.705**	**85.099**	82.311	79.188
+SSPT-PN	91.599	84.939	**82.337**	**79.322**
$\text{BERT}_{\text{LARGE}}$	90.970	84.201	81.504	78.413
+SSPT	**92.747**	**86.859**	**85.029**	**82.069**

Table 4: Results on SQuAD

Method	Short Ans F1	Long Ans F1
$\text{BERT}_{\text{BASE}}$	47.27	61.02
$\text{BERT}_{\text{BASE}}$+SSPT	**50.40**	**63.35**
$\text{BERT}_{\text{LARGE}}$	52.7	64.7
$\text{BERT}_{\text{LARGE}}$+SSPT	**54.2**	**65.85**

Table 5: Dev Set Results on Natural Questions

Method	Facts		Answer	
	F1	Exact	F1	Exact
$\text{BERT}_{\text{BASE}}$	84.00	53.15	73.86	59.97
$\text{BERT}_{\text{BASE}}$+SSPT	**85.13**	**56.58**	**77.25**	**63.31**
$\text{BERT}_{\text{LARGE}}$	85.27	55.99	75.48	61.62
$\text{BERT}_{\text{LARGE}}$+SSPT	**86.17**	**57.57**	**79.39**	**65.87**

Table 6: Results on HotpotQA

Method	**MRPC**	**QQP**
	Accuracy	Accuracy
$\text{BERT}_{\text{BASE}}$	84.80	90.96
$\text{BERT}_{\text{BASE}}$+SSPT	**87.75**	91.18
$\text{BERT}_{\text{LARGE}}$	85.54	91.24
$\text{BERT}_{\text{LARGE}}$+SSPT	86.77	91.55

Table 7: Results on MRPC and QQP

6.1 SQuAD

Relative to $\text{BERT}_{\text{BASE}}$ we find a 3 point improvement in F1 for SQuAD 1.1 and a nearly 6 point improvement for SQuAD 2.0. In terms of error rate reduction the improvement is similar, 28% and 25% respectively. The error rate reduction for $\text{BERT}_{\text{LARGE}}$ is 20% and 19% for SQuAD 1.1 and 2.0 respectively.

In reading comprehension tasks, the pointer network for answer selection is pre-trained through the span selection task. We measure how much of the improvement is due to this final layer pre-training versus the extended pre-training for the transformer encoder layers by discarding the pre-trained pointer network and randomly initializing. This configuration is indicated as $\text{BERT}_{\text{BASE}}$+SSPT-PN. Surprisingly, the pre-training of the pointer network is not a significant factor in the improved performance on reading comprehension, indicating the improvement is instead coming through a better language understanding in the transformer.

Figure 4 shows the improvement from SSPT on SQuAD 1.1 and 2.0 as the amount of training data increases. While there is significant improvement at 100% training, the improvement is even more pronounced with less training data. We hypothesize that this is due to the close connection of span selection pre-training with reading comprehension. This effect is strongest for SQuAD 1.1, which like span selection pre-training always contains a correct answer span in the passage.

6.2 Natural Questions

The work of Alberti et al. [2019a], which gets the $\text{BERT}_{\text{LARGE}}$ performance listed in Table 5, is the highest ranking single model submission that does not use data augmentation with a published paper. Our implementation of

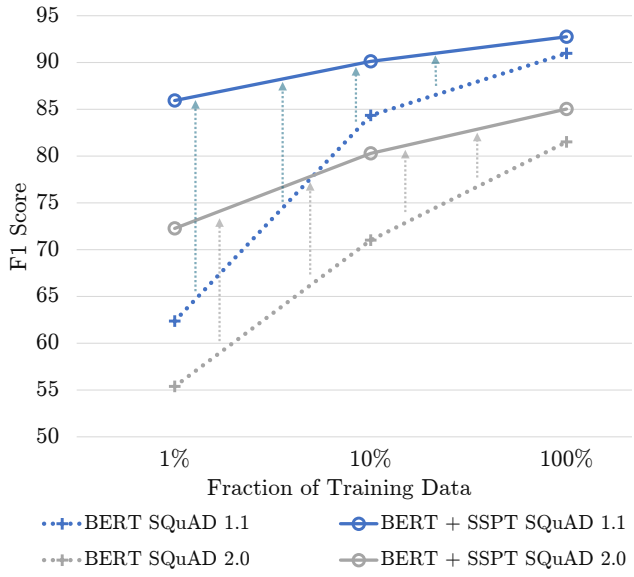

Figure 4: Learning curve improvement for $\text{BERT}_{\text{LARGE}}$ with SSPT

$\text{BERT}_{\text{LARGE}}$+SSPT, therefore, provides a 1.5% improvement over the best BERT-for-QA model performance that we are aware of on the NQ data set. In future work, we intend to explore data augmentation on top of $\text{BERT}_{\text{LARGE}}$+SSPT for further improvements.

6.3 HotpotQA

In HotpotQA, unlike the other QA datasets, multiple passages are provided. We use the BERT transformer in two places, for supporting fact prediction to build the representations of each passage, and in answer prediction as in the other QA tasks. We find the most substantial gains of almost 4 F1 points for answer selection, the QA task most similar to span selection pre-training. Interestingly, we also find improvement of almost one point F1 in supporting fact prediction, demonstrating that the learned representations can generalize well to multiple QA sub-tasks.

HotpotQA also comes with its own leaderboard (https://hotpotqa.github.io/). A good number of submissions on this leaderboard are based on $\text{BERT}_{\text{BASE}}$ or $\text{BERT}_{\text{LARGE}}$. We made an initial submission to this leaderboard, called TAP, which occupied Rank-5 at the time of submission and the underlying architecture employed $\text{BERT}_{\text{BASE}}$. Next, we replaced $\text{BERT}_{\text{BASE}}$ with $\text{BERT}_{\text{LARGE}}$+SSPT, calling that model TAP-2. This change resulted in a 7.22% absolute gain in the Joint $F1$ score. TAP-2 (single model) occupies Rank-2 on the leaderboard. An ensemble version of TAP-2 further offered a gain of 1.53% pushing the SSPT augmented TAP-2 (ensemble) on the top of the HotpotQA leaderboard while pushing the original submission down to Rank-7.

6.4 Comparison to Previous Work

We also compare our span selection pre-training data with the data distributed by Dhingra et al. [2018]. This data consists of approximately 2 million instances constructed using the abstract and body structure of Wikipedia. In contrast, our approach to pre-training can generate data in unlimited quantity from any text source without assuming a particular document structure. When only one million training steps are used, both sources of pre-training are equally effective. But when moving to ten million steps of training, our data produces models that give over one percent better F1 on both SQuAD 1.1 and 2.0. This suggests the greater quantity of data possible through SSPT is a powerful advantage.

7 Conclusion and Future Work

Span selection pre-training is effective in improving reading comprehension across four diverse datasets, including both generated and natural questions, and with provided contexts of passages, documents and even passage sets. We also find some effectiveness in improving paraphrase detection. This style of pre-training focuses the model on finding semantic connections between two sequences, and supports a style of cloze that can train deep semantic understanding

without demanding memorization of world knowledge in the model. The span selection task is suitable for pre-training on any domain, since it makes no assumptions about document structure or availability of summary/article pairs. This allows pre-training of language understanding models in a very generalizable way.

In future work, we will address end-to-end question answering with pre-training for both the answer selection and retrieval components. We hope to progress to a model of general purpose language modeling that uses an indexed long term memory to retrieve world knowledge, rather than holding it in the densely activated transformer encoder layers.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. *CoRR*, abs/1906.05416, 2019a. URL http://arxiv.org/abs/1906.05416.

Chris Alberti, Kenton Lee, and Michael Collins. A bert baseline for the natural questions, 2019b. URL https://arxiv.org/abs/1901.08634v2.

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. URL https://arxiv.org/pdf/1810.04805.pdf.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. Simple and effective semi-supervised question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 582–587, 2018. URL https://aclweb.org/anthology/N18-2092.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1031.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*, 2019.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *TACL*, 2019. URL https://tomkwiat.users.x20web.corp.google.com/papers/natural-questions/main-1455-kwiatkowski.pdf.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, 2016. URL https://aclweb.org/anthology/D16-1241.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1202. URL http://aclweb.org/anthology/N18-1202.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Preprint*, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016. URL https://aclweb.org/anthology/D16-1264.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL http://dx.doi.org/10.1561/1500000019.

Mrinmaya Sachan and Eric Xing. Self-training for jointly learning to ask and answer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1058. URL http://aclweb.org/anthology/N18-1058.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *CoRR*, abs/1701.06538, 2017. URL http://arxiv.org/abs/1701.06538.

Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*, 2019.

Wilson L Taylor. "Cloze procedure": A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433, 1953.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics, 2018. URL https://aclweb.org/anthology/W18-5446.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019. URL http://arxiv.org/abs/1906.08237.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.