Pre-training Transformer Models with Sentence-Level Objectives for Answer Sentence Selection

Luca Di Liello[1*], **Siddhant Garg**[2], **Luca Soldaini**[3†], **Alessandro Moschitti**[2]
[1]University of Trento , [2]Amazon Alexa AI, [3]Allen Institute for AI
luca.diliello@unitn.it
{sidgarg,amosch}@amazon.com
lucas@allenai.org

Abstract

An important task for designing QA systems is answer sentence selection (AS2): selecting the sentence containing (or constituting) the answer to a question from a set of retrieved relevant documents. In this paper, we propose three novel sentence-level transformer pre-training objectives that incorporate paragraph-level semantics within and across documents, to improve the performance of transformers for AS2, and mitigate the requirement of large labeled datasets. Specifically, the model is tasked to predict whether: (i) two sentences are extracted from the same paragraph, (ii) a given sentence is extracted from a given paragraph, and (iii) two paragraphs are extracted from the same document. Our experiments on three public and one industrial AS2 datasets demonstrate the empirical superiority of our pre-trained transformers over baseline models such as RoBERTa and ELECTRA for AS2.

1 Introduction

Question Answering (QA) finds itself at the core of several commercial applications, for e.g., virtual assistants such as Google Home, Alexa and Siri. Answer Sentence Selection (AS2) is an important task for QA Systems operating on unstructured text such as web documents. When presented with a set of relevant documents for a question (retrieved from a web index), AS2 aims to find the best answer sentence for the question.

The recent popularity of pre-trained transformers (Devlin et al., 2019; Liu et al., 2019; Clark et al., 2020), has made them the de-facto approach for most QA tasks, including AS2. Several research works (Garg et al., 2020; Laskar et al., 2020; Lauriola and Moschitti, 2021) fine-tune transformers for AS2, by posing it as a sentence-pair task and performing inference over the encoded representations of the question and answer candidates.

AS2 is a knowledge-intensive complex reasoning task, where the answer candidates for a question can stem from multiple documents, possibly on different topics linked to concepts in the question. While there have been recent works (Ginzburg et al., 2021; Caciularu et al., 2021) proposing pre-training strategies for obtaining multi-document aware document representations over long input encoders such as the Longformer (Beltagy et al., 2020), there has been limited research (Giorgi et al., 2021) on enhancing sentence-pair representations with paragraph and document level semantics.

Furthermore, obtaining high quality human labeled examples for AS2 is expensive and time consuming, due to the large number of answer candidates to be annotated for each question. Domain-specific AS2 datasets such as WikiQA (Yang et al., 2015) and TREC-QA (Wang et al., 2007) only contain a few thousand questions. Garg et al. (2020) show that effectively fine-tuning pre-trained transformers on these domain specific AS2 datasets requires an intermediate fine-tuning transfer on a large scale AS2 dataset (ASNQ).

Towards improving the downstream performance of pre-trained transformers for AS2 and mitigating the requirement of large scale labeled data for fine-tuning, we propose three novel sentence-level transformer pre-training objectives, which can incorporate paragraph-level semantics across multiple documents. Analogous to the sentence-pair nature of AS2, we design our pre-training objectives to operate over a pair of input text sequences. The model is tasked with predicting: (i) whether the sequences are two sentences extracted from the same paragraph, (ii) whether the first sequence is a sentence that is extracted from the second sequence (paragraph), and (iii) whether the sequences are two paragraphs belonging to the same document.

We evaluate our paragraph-aware pre-trained transformers for AS2 on three popular public datasets: ASNQ, WikiQA and TREC-QA; and one

[*]Work done as an intern at Amazon Alexa AI
[†]Work completed at Amazon Alexa AI

industrial QA benchmark [1]. Results show that our pre-training can improve the performance of fine-tuning baseline transformers such as RoBERTa and ELECTRA on AS2 by $\sim 3-4\%$ points without requiring any additional data (labeled/unlabeled).

2 Related Work

Answer Sentence Selection (AS2) Earlier approaches for AS2 used CNNs (Severyn and Moschitti, 2015) or alignment networks (Shen et al., 2017a; Tran et al., 2018) to learn and score question and answer representations. Since then, compare-and-aggregate architectures have also been extensively studied (Wang and Jiang, 2017; Bian et al., 2017; Yoon et al., 2019). Garg et al. achieved state-of-the-art results by fine-tuning transformers on a large QA corpora first, and then adapting to a smaller AS2 dataset.

Token-Level Pre-training Objectives Masked Language Modeling (MLM) is one of the most popular token-level pre-training objectives used for transformers (Devlin et al., 2019; Liu et al., 2019). Some other models trained using token-level pre-training objectives are Yang et al. (2020) and Clark et al. (2020). Joshi et al. (2020) modify MLM to a span-prediction objective to make the model generalize well to machine reading tasks in QA.

Sentence-Level Pre-training Objectives In addition to MLM, Devlin et al. (2019) uses the next sentence prediction (NSP) objective, which was later shown to not provide empirically improvements over MLM by Liu et al. (possibly due to the task being very simple). Lan et al. (2020) propose a sentence order prediction (SOP) objective. Ippolito et al. (2020) enhance NSP to a multiple-choice prediction of the next sentence over a set of candidates, however they embed each sentence independently without cross-attention between them similar to (Reimers and Gurevych, 2019). Gao et al. (2021) propose a supervised contrastive learning approach for enhancing sentence representations for textual similarity tasks.

Paragraph/Document-level Semantics (Chang et al., 2019) pre-train Bi-HLSTMs for obtaining hierarchical document representations. HIBERT (Zhang et al., 2019) uses document-level token masking and sentence masking pre-training objectives for generative tasks such as document summarization. Transformer pre-training objectives at different granularities of document semantics are discussed in (Li et al., 2020) for fact verification, and in (Chang et al., 2020) for retrieval. Ginzburg et al.; Caciularu et al. propose pre-training strategies for document embeddings for retrieval tasks such as document-matching. DeCLUTR (Giorgi et al., 2021) uses contrastive learning for cross-encoding two sentences coming from the same/different documents in a transformer, and is evaluated on pairwise binary classification tasks like natural language inference. Our work differs from this since we use a cross-encoder architecture to capture cross-attention between the question and answer, and evaluate our approach on the relevance ranking task of AS2 over hundreds of candidates. Contemporary works (Di Liello et al., 2022) pre-train transformers using paragraph-aware objectives for multi-sentence inference tasks. Our work differs from this since we only encode a pair of sentences using the transformer, while the former encode multiple sentences and use sophisticated prediction heads to aggregate information across multiple representations.

Transformers for Long Inputs Longformer (Beltagy et al., 2020), Big Bird (Zaheer et al., 2020), etc. model very long inputs (e.g, entire documents) by reducing the complexity of transformer attention. This provides longer context, which is useful for machine reading and summarization.

3 Answer Sentence Selection (AS2)

In this section we formally define the task of AS2. Given a question q and a set of answer candidates $A=\{a_1, \ldots, a_n\}$, the objective is to select the candidate $\bar{a} \in A$ that best answers q. AS2 can be modeled as a ranking task over A to learn a scoring function $f : Q \times A \rightarrow \mathbb{R}$ that predicts the probability $f(q, a)$ of an answer candidate a being correct. The best answer $\bar{a}$ corresponds to $\text{argmax}_{i=1}^{n} f(q, a_i)$. Pre-trained transformers are used as QA pair encoders for AS2 to approximate the function f.

4 Sentence-Level Pre-training Objectives

Documents are typically organized into paragraphs, by humans, to address the document's general topic from different viewpoints. We propose three pre-training objectives to exploit the intrinsic information contained in the structure of documents. For all these objectives, we provide a pair of text sequences as input to the transformer to jointly reason over them, analogous to the AS2 task.

[1] We will release code and pre-trained models at https://github.com/amazon-research/wqa-pretraining

Spans in Same Paragraph (SSP) Given two sequences (A, B) as input to the transformer, the objective is to predict if A and B belong to the same paragraph in a document. To create positive pairs (A, B), given a document D, we extract two small, contiguous and disjoint subsets of sentences to be used as A and B from a single paragraph $P_i \in D$. To create negative pairs, we sample spans of sentences B' from different paragraphs $P_j, j \neq i$ in the same document D (hard negatives) and also from different documents (easy negatives). The negative pairs correspond to (A, B'). Posing the above pre-training objective in terms of spans (instead of sentences) allows for modifying the lengths of the inputs A, B (by changing number of sentences $\in A, B$). When fine-tuning transformers for AS2, typically the question is provided as the first input and a *longer* answer candidate/paragraph is provided as the second input. For our experiments (Sec 5), we use a longer span for input B than A.

Span in Paragraph (SP) Given two sequences (A, B) as input to the transformer, the objective is to predict if A is a span of text extracted from a paragraph B in a document. To create positive pairs (A, B), given a paragraph P_i in a document D, we extract a small contiguous span of sentences A from it and create the input pair as $(A, P_i \setminus A)$. To create negative pairs, we select other paragraphs $P_j, j \neq i$ in the same document D and remove a randomly chosen span A' from each of them. The negative pairs correspond to $(A, P_j \setminus A')$. This is necessary to ensure that the model does not simply recognize whether the second input is a complete paragraph or a clipped version. To create easy negatives, we use the above approach for paragraphs P_j sampled from documents other than D.

Paragraphs in Same Document (PSD) Given two sequences (A, B) as input to the transformer, the objective is to predict if A and B are paragraphs belonging to the same document. To create positive pairs (A, B), given a document D_k, we randomly select paragraphs $P_i, P_j \in D_k$ and obtain a pair (P_i, P_j). To create negative pairs, we randomly select $P'_j \notin D_k$, and obtain a pair (P_i, P'_j).

5 Experiments

5.1 Datasets

Pre-training To eliminate any improvements stemming from the usage of more data, we perform pre-training on the same corpora as RoBERTa: English Wikipedia, the BookCorpus, OpenWebText and CC-News. We perform continuous pre-training starting from RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2020) checkpoints, using a combination of our objectives with the original ones (MLM for RoBERTa and MLM + Token Detection for ELECTRA). Refer to Appendix A for complete details.

AS2 Fine-tuning We consider three public and one industrial AS2 benchmark as fine-tuning datasets for AS2 (statistics presented in Appendix A). We use standard evaluation metrics for AS2: Mean Average Precision (MAP), Mean Reciprocal Recall (MRR) and Precision@1 (P@1).

• **ASNQ** is a large-scale AS2 dataset (Garg et al., 2020) with questions from Google search engine queries, and answer candidates extracted from a Wikipedia page. ASNQ is a modified version of the Natural Questions (NQ) (Kwiatkowski et al., 2019), obtained by labeling sentences from long answers that contain the short answer as positives and all others as negatives. We use the dev and test splits released by Soldaini and Moschitti[2].

• **WikiQA** is a popular AS2 dataset (Yang et al., 2015) where questions are derived from query logs of the Bing search engine, and the answer candidates are extracted from a Wikipedia page. This dataset has a subset of questions having no correct answers (*all-*) or having only correct answers (*all+*). We remove both the (*all-*) and (*all+*) questions for our experiments (standard "clean" setting).

• **TREC-QA** is a popular AS2 dataset (Wang et al., 2007) of factoid questions, extracted from the TREC-8 to TREC-13 QA tracks. The answer candidates are sentences that contain one or more non-stopwords in common with the question, extracted from multiple documents. For the dev and test sets, we remove questions without answers, or having only correct or only incorrect answer candidates ("clean" setting (Shen et al., 2017b)).

• **WQA** A large scale industrial AS2 dataset containing *non-representative de-identified* user questions from Alexa virtual assistant. For every question, ∼15 answer candidates are collected from a large web index of more than 100M documents using Elasticsearch. Results on WQA are presented relative to the RoBERTa-Base baseline due to the data being internal.

[2] https://github.com/alexa/wqa-cascade-transformers

Model	ASNQ			WikiQA			TREC-QA			WQA		
	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR
RoBERTa-Base	61.8 (0.2)	66.9 (0.1)	73.1 (0.1)	78.3 (2.8)	85.8 (1.3)	87.2 (1.3)	90.0 (1.9)	89.7 (0.7)	94.4 (1.1)		Baseline	
(Ours) RoBERTa + SSP	**64.1** (0.3)	68.1 (0.2)	**74.5** (0.3)	**82.9** (0.7)	**88.7** (0.3)	**89.9** (0.4)	88.5 (1.2)	89.3 (0.7)	93.6 (0.6)	+0.2%	+0.6%	+0.3%
(Ours) RoBERTa + SP	**64.1** (0.2)	**68.3** (0.1)	**74.5** (0.2)	81.0 (0.8)	87.7 (0.3)	88.9 (0.4)	**90.9** (2.6)	90.1 (0.8)	94.7 (1.3)	+0.4%	+0.7%	+0.5%
(Ours) RoBERTa + PSD	62.6 (0.4)	67.7 (0.2)	73.7 (0.3)	80.5 (1.6)	86.4 (1.1)	88.0 (1.0)	90.3 (1.3)	**90.3** (0.5)	**95.1** (0.7)	+0.4%	+0.7%	+0.5%
(Ours) RoBERTa + All	63.9 (0.4)	68.0 (0.1)	74.1 (0.2)	82.5 (0.9)	88.2 (0.4)	89.5 (0.4)	87.9 (1.2)	89.3 (0.7)	93.4 (0.6)	**+0.5%**	**+0.8%**	**+0.6%**
TANDA RoBERTa	-	-	-	83.0 (1.3)	88.5 (0.8)	89.9 (0.8)	89.7 (0.0)	90.1 (0.6)	94.1 (0.4)	+0.5%	+0.5%	+0.5%
ELECTRA-Base	62.4 (0.4)	67.5 (0.2)	73.6 (0.2)	77.1 (4.0)	85.0 (2.6)	86.5 (2.7)	90.3 (1.7)	89.9 (0.4)	94.0 (0.9)	+1.0%	+1.2%	+0.9%
(Ours) ELECTRA + SSP	**65.3** (0.3)	**69.7** (0.2)	**75.7** (0.2)	**82.5** (2.0)	**88.6** (1.4)	**90.0** (1.4)	88.5 (1.9)	89.6 (0.7)	93.5 (0.9)	+1.4%	+1.5%	+1.3%
(Ours) ELECTRA + SP	65.0 (0.2)	69.0 (0.1)	75.1 (0.1)	81.8 (2.3)	88.1 (1.5)	89.5 (1.5)	91.2 (1.5)	90.3 (0.7)	94.6 (0.7)	+1.4%	+1.5%	+1.3%
(Ours) ELECTRA + PSD	**65.3** (0.4)	68.9 (0.3)	75.1 (0.3)	78.6 (0.7)	85.6 (0.7)	87.3 (0.6)	85.9 (2.2)	87.9 (1.1)	92.2 (1.1)	**+1.6%**	**+1.6%**	+1.3%
(Ours) ELECTRA + All	65.0 (0.3)	69.3 (0.2)	75.2 (0.2)	80.8 (1.9)	87.3 (1.2)	88.7 (1.1)	**92.6** (1.8)	**90.4** (0.4)	**95.5** (1.0)	+1.5%	**+1.6%**	**+1.4%**
TANDA ELECTRA	-	-	-	85.6 (1.1)	90.2 (0.8)	91.4 (0.7)	92.6 (1.5)	91.6 (0.7)	95.5 (0.7)	+1.9%	+1.6%	+1.5%

Table 1: Results (with std. dev. across 5 runs in parentheses) of our pretrained transformers when fine-tuned on AS2 datasets. SSP, SP, PSD denote our pretraining objectives, and 'All' denotes using SSP+SP+PSD together. TANDA uses **additional labeled data** as an intermediate transfer step. We underline statistically significant improvements over the baseline (T-test at a 95% confidence level). Results on WQA are relative to the RoBERTa baseline.

5.2 Experimental Setup and Details

We use our 3 pre-training objectives: SSP, SP and PSD, for both RoBERTa and ELECTRA, obtaining 6 different continuously pre-trained models. We set the maximum pre-training steps to 400k for SSP and 200k for SP and PSD. This corresponds to each model processing $\sim 210B$ tokens during pre-training, which is about 10% of the $\sim 2100B$ tokens used for pre-training RoBERTa. Notice also that the compute FLOPs are even less than the 10% of the original training because we used a shorted max sequence length. More details about the continuous pre-training hyper-parameters are given in Appendix B.

We also combine all 3 objectives together (SSP+SP+PSD) for both RoBERTa and ELECTRA, with the same setting as SSP. We fine-tune each of our pre-trained models on all four AS2 datasets (with early stopping on the dev set) and compute results on their respective test splits.

Baselines We use RoBERTa and ELECTRA models as baselines. We also use TANDA (Garg et al., 2020), the state of the art for AS2, as an upper-bound baseline as it uses an additional intermediate transfer step on ASNQ ($\sim 20M$ labeled QA pairs). Note that we don't consider Ginzburg et al.; Caciularu et al.; Chang et al. as baselines as they are designed for document-matching and retrieval tasks, and Beltagy et al.; Zaheer et al. as they are used for long-context tasks like MR and summarization.

5.3 Results

We present results of our pre-trained models on the AS2 datasets in Table 1. We observe that the models trained with our pre-training objectives significantly outperform the baseline models when fine-tuned for the AS2 tasks. For example, on ASNQ, using our SP objective with RoBERTa-Base gains 2.3% in P@1 over the baseline RoBERTa-Base model. On WikiQA, the performance gap is even larger with the SSP objective corresponding to 4.6% points for RoBERTa-Base and 5.4% for ELECTRA-Base over the corresponding baselines. Performance improvements on TREC-QA and WQA are smaller but consistent, around 1% and 0.6% in P@1. Combining SSP+SP+PSD together consistently achieves either the best results (TREC-QA and WQA), or close to the best results (ASNQ and WikiQA).

For questions in ASNQ and WikiQA, all candidate answers are extracted from a *single* Wikipedia document, while for TREC-QA and WQA, candidate answers come from *multiple* documents extracted from heterogeneous web sources. By design of our objectives SSP, SP and PSD, they perform differently when fine-tuning on different datasets. For example, SSP aligns well with ASNQ and WikiQA as they contain many negative candidates, per question, extracted from the same document as the positive (i.e, 'hard' negatives). As per our design of the SSP objective, for every positive sequence pair, we sample 2 'hard' negatives coming from the same document as the positive pair. The presence of hard negatives is of particular importance for WikiQA and ASNQ, as it forces the models to learn and contrast more subtle differences between answer candidates, which might likely be more related as they come from the same document.

On the other hand, PSD is designed so as to see paragraphs from same or different documents (with no analogous concept of 'hard' negatives of SSP and SP). For this reason, PSD is better aligned for fine-tuning on datasets where candidates are extracted from multiple documents, such as WQA

Model+Data Sampling	ASNQ	WikiQA	TREC-QA	WQA
RoBERTa-Base	61.8	78.3	90.0	Baseline
+ SSP Data (**MLM-only**)	63.4	76.7	87.4	-0.6%
+ SSP	**64.1**	**82.9**	**88.5**	**+0.2**%
+ SP Data (**MLM-only**)	62.8	76.8	88.8	-1.0%
+ SP	**64.1**	**81.0**	**90.9**	**+0.4**%
+ PSD Data (**MLM-only**)	**64.1**	79.1	87.1	-1.3%
+ PSD	62.6	**80.5**	**90.3**	**+0.4**%

Table 2: P@1 of our pretrained models using SSP, SP and PSD objectives in addition to only MLM. We highlight in bold and underline results like in Table 1.

Model + Pre-training Objective	Accuracy	F1
RoBERTa-Base + **SSP**	91.8	83.1
ELECTRA-Base + **SSP**	90.4	79.9
RoBERTa-Base + **SP**	91.3	83.3
ELECTRA-Base + **SP**	89.9	80.1
RoBERTa-Base + **PSD**	83.5	61.4
ELECTRA-Base + **PSD**	82.3	57.1
BERT (Devlin et al., 2019) (NSP)	96.9	97.1
ALBERT (Lan et al., 2020) (SOP)	93.7	94.7

Table 3: Comparison of accuracy and F1-score of pre-training objectives on the pre-training validation set.

and TREC-QA.

Comparison with TANDA For RoBERTa, our pre-trained models can surprisingly improve/achieve comparable performance to TANDA. Note that our models achieve this performance without using the latter's additional $\sim 20M$ labeled ASNQ QA pairs. This lends support to our pre-training objectives mitigating the requirement of large scale labeled data for AS2 fine-tuning. For ELECTRA, we only observe comparable performance to TANDA for WQA and TREC-QA.

Ablation: MLM-only Pre-training To mitigate any improvements stemming from the specific data sampling techniques used by our objectives, we pre-train 3 models (starting from RoBERTa-Base) with the same data sampling as each of the SSP, SP and PSD models, but only using the MLM objective.

We report results in Table 2, and observe that, almost always, models pre-trained only with MLM under-perform models trained with SSP, SP and PSD objectives in addition to MLM. Thus, the empirical improvements of our methods are derived from the novel pre-training objectives, and not data sampling. Surprisingly, for some models, the MLM-only continuous pre-training performs worse than the baseline RoBERTa-Base. We believe that restarting the training with a different learning-rate[3], a shorter sequence length, and without the original optimizer and scheduler internal states (for a small amount of steps) is sub-optimal for the model.

Ablation: Pre-training Task 'Difficulty' We evaluate the pre-trained models (after convergence) on their specific tasks over the validation split of Wikipedia (to enable evaluating baselines such as BERT and ALBERT). Table 3 summarizes the accuracy and F1 of the models on the various tasks.

The results show that our objectives are generally *harder* than NSP (Next Sentence Prediction by Devlin et al., 2019) and SOP (Sentence Order Prediction by Lan et al., 2020). In fact, NSP and SOP have been shown to not add any significant performance improvements in addition to MLM (Liu et al., 2019), and this corresponds to the model being able to perform this task extremely well (dev accuracy $\sim 94\%$ with NSP and $\sim 97\%$ with SOP) without learning any new semantics that may be useful for downstream tasks.

On the other hand, our pre-training objectives are "more challenging" than these previously proposed objectives due to the requirement of reasoning over multiple paragraphs and multiple documents, addressing same or different topics at the same time. In fact, Table 3 shows that after convergence, our pre-trained model still finds it *difficult* to achieve a higher accuracy for our sentence level pre-training tasks. Empirically in Table 1, we observed that pre-training with our objectives is able to rank the more relevant answers at the top, which we hypothesize is due to the model learning how to reason over multiple paragraphs and documents already while performing continuous pre-training.

6 Conclusion

In this paper we have presented three sentence-level pre-training objectives for transformers to incorporate paragraph and document-level semantics. Our objectives predict whether (i) two sequences are sentences extracted from the same paragraph, (ii) first sequence is a sentence extracted from the second, and (iii) two sequences are paragraphs belonging to the same document. We evaluate our pre-trained models for the task of AS2 on four datasets. Our results show that our pre-trained models outperform the baseline transformers such as RoBERTa and ELECTRA.

[3]The original models use a triangular learning-rate

Limitations

We only consider English language datasets for our experiments in this paper. However we hypothesize that our pre-training objectives should provide similar performance improvements when extended to other languages with limited morphology, like English. The pre-training objectives proposed in our work are designed considering Answer Sentence Selection (AS2) as the target task, and can be extended for other tasks like Natural Language Inference, Question-Question Similarity, etc. in future work. The pre-training experiments in our paper require large amounts of GPU and compute resources (multiple NVIDIA A100 GPUs running for several days) to finish the model pre-training. This makes re-training models using our pre-training approaches computationally expensive using newer data. To mitigate this, we are releasing our code and pre-trained model checkpoints at `https://github.com/amazon-research/wqa-pretraining`, which can directly be used by fine-tuning them on AS2 datasets.

Acknowledgements

We thank the anonymous reviewers and the ARR action-editor for their valuable suggestions. We would like to thank Thuy Vu for developing and sharing the WQA dataset.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew Peters, Arie Cattan, and Ido Dagan. 2021. CDLM: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ming-Wei Chang, Kristina Toutanova, Kenton Lee, and Jacob Devlin. 2019. Language model pre-training for hierarchical document representations. *CoRR*, abs/1901.09128.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. 2022. Torchmetrics - measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022. Paragraph-based transformer pre-training for multi-sentence inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2521–2531, Seattle, United States. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siddhant Garg and Alessandro Moschitti. 2021. Will this question be answered? question filtering via answer model distillation for efficient question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7329–7346, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7780–7788.

Dvir Ginzburg, Itzik Malkiel, Oren Barkan, Avi Caciularu, and Noam Koenigstein. 2021. Self-supervised document similarity ranking via contextualized language models and hierarchical inference. In *Findings of the Association for Computational Linguis-*

tics: ACL-IJCNLP 2021, pages 3088–3098, Online. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Daphne Ippolito, David Grangier, Douglas Eck, and Chris Callison-Burch. 2020. Toward better storylines with sentence-level language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7472–7478, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Md Tahmid Rahman Laskar, Jimmy Xiangji Huang, and Enamul Hoque. 2020. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5505–5514, Marseille, France. European Language Resources Association.

Ivano Lauriola and Alessandro Moschitti. 2021. Answer sentence selection using local and global context in transformer models. ECIR.

Xiangci Li, Gully A. Burns, and Nanyun Peng. 2020. A paragraph-level multi-task learning model for scientific fact-verification. *CoRR*, abs/2012.14500.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017a. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017b. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.

Luca Soldaini and Alessandro Moschitti. 2020. The cascade transformer: an application for efficient answer sentence selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5697–5708, Online. Association for Computational Linguistics.

Quan Hung Tran, Tuan Lai, Gholamreza Haffari, Ingrid Zukerman, Trung Bui, and Hung Bui. 2018. The context-dependent additive recurrent neural net. 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 1274–1283, New Orleans, Louisiana. Association for Computational Linguistics.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yi Yang, Scott Wen-tau Yih, and Chris Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. ACL - Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020.

Xlnet: Generalized autoregressive pretraining for language understanding.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Appendix

A Datasets

A.1 Pre-training

For continued pre-training, we pre-process the English Wikipedia [4], the BookCorpus[5], OpenWebText (Gokaslan and Cohen, 2019) and the CC-News[6] datasets. We do not use the STORIES dataset as it is no longer available for research use [7]. We clean every dataset by removing headers, titles, tables and any HTML content. For every document, we keep paragraphs containing at least 60 characters and documents containing at least 200 characters. After cleaning, we obtain 5GB, 10GB, 34GB and 360GB of raw text from the BookCorpus, Wikipedia, OpenWebText and CC-News respectively. We split paragraph into lists of sentences using the blingfire tokenizer[8]. We present the details of our pre-training objectives in Section 4. We present the details on sampling lengths and number of negatives for each of the objectives below:

• **Spans in Same Paragraph (SSP)** We randomly sample the number of sentences in A in the interval $[1, 3]$ and B in $[1, 5]$. This is to keep the inputs to the model analogous to those in AS2 (shorter question text, followed by longer answer text). We sample up to 2 hard negatives from the same paragraph as A (if possible), and sample easy negatives from other documents so as to make the total number of negatives to be 4.

• **Span in Paragraph (SP)** We randomly sample the number of sentences in $A \in P_i$ in the interval $[1, 3]$. The number of sentences in the right part is given by the length of $P_i \setminus A$ (positive pair) or $P_j \setminus X_j$ (negative pair). Similar to SSP, we sample up to 2 hard negatives from the same document (if possible), and sample easy negatives from other documents so as to make the total number to be 4.

• **Paragraphs in Same Document (PSD)** We chose a random pair of paragraphs A and B from a single document and then we randomly sample 4 paragraphs from other documents to create the negative pairs with A.

A.2 Fine-tuning

Here we present statistics and links for downloading the AS2 datasets used: ASNQ[9], WikiQA[10], TREC-QA and WQA; to benchmark our pre-trained models. Table 4 shows the number of unique questions and answer candidates for each dataset and for each split.

Dataset	Split	# Q	# C	Avg. # C/Q
ASNQ	Train	57,242	20,377,568	356.0
	Dev	1,336	463,914	347.2
	Test	1,336	466,148	348.9
WikiQA	Train	2,118	20,360	9.6
	Dev	122	1,126	9.2
	Test	237	2,341	9.9
TREC-QA	Train	1,226	53,417	43.6
	Dev	69	1,343	19.5
	Test	68	1,442	21.2
WQA	Train	9,984	149,513	15.0
	Dev	5,000	74,805	15.0
	Test	5,000	74,712	14.9

Table 4: Data Statistics for AS2 dataset. "Avg. # C/Q" is the average number of answer candidates per question.

B Experimental Setup

We experiment with the *base* architecture, which uses an hidden size of 768, 12 transformer layers, 12 attention heads and feed-forward size of 3072.

Pre-training We perform continued pre-training starting from the publicly released checkpoints of RoBERTa-Base (Liu et al., 2019) and ELECTRA-Base (Clark et al., 2020). We optimize using Adam, which we instantiate with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We use a triangular learning rate with 10k warmup steps. The peak learning rate is set to $1 * 10^{-4}$. We apply a weight decay of 0.01, gradient clipping when values are larger than 1.0 and dropout ratio is set to 0.1. We set the batch size to 4096 examples for every combination of models and objectives. We truncate the input sequences to 128 tokens for SSP and to 256 tokens with SP and PSD. Finally, we perform 400k training steps with models using SSP and 200k steps with the other objectives: SP and PSD. The total amount of tokens seen in the continued pre-training is the same for all models and equal to $\sim 210B$.

We combine the binary classification loss of SSP, SP and PSD with MLM for RoBERTa and with MLM (of the generator) and TD (token detection)

[4] https://dumps.wikimedia.org/enwiki/20211101/

[5] https://huggingface.co/datasets/bookcorpusopen

[6] https://commoncrawl.org/2016/10/news-dataset-available/

[7] https://github.com/tensorflow/models/tree/archive/research/lm_commonsense#1-download-data-files

[8] https://github.com/microsoft/BlingFire

[9] https://github.com/alexa/wqa_tanda

[10] http://aka.ms/WikiQA

for ELECTRA. For RoBERTa, we perform binary classification on the first [CLS] token in addition to MLM. For ELECTRA, using the generator + discriminator architecture, we perform MLM on the generator; and token-detection along with binary classification on the discriminator using our pre-training objectives. Through experimentation, for RoBERTa, we use equal weights for MLM and our pre-training objectives. For ELECTRA, we combine MLM, TD and our pre-training objectives with the weights 1.0, 50.0 and 1.0 respectively.

Fine-tuning The evaluation of the models is performed on four different datasets for Answer Sentence Selection. We maintain the same hyperparameters used in pre-training apart from the learning rate, the number of warmup steps and the batch size. We do early stopping on the development set if the number of non-improving validations (patience) is higher than 5. For ASNQ, we found that using a very large batch size is beneficial, providing a higher accuracy. We use a batch size of 2048 examples on ASNQ for RoBERTa models and 1024 for ELECTRA models. The peak learning rate is set to $1 * 10^{-5}$ for all models, and the number of warmup steps to 1000. For WikiQA, TREC-QA and WQA, we select the best batch size out of $\{16, 32, 64\}$ and learning rate out of $\{2*10^{-6}, 5*10^{-5}, 1*10^{-5}, 2*10^{-5}\}$ using cross-validation. We train the model for 6 epochs on ASNQ, and up to 40 epochs on WikiQA, TREC-QA, and WQA. The performance of practical AS2 systems is typically measured using Precision-at-1 P@1 (Garg and Moschitti, 2021). In addition to P@1, we also use Mean Average Precision (MAP) and Mean Reciprocal Recall (MRR) to evaluate the ranking of the set of candidates produced by the model.

We used metrics from Torchmetrics (Detlefsen et al., 2022) to compute MAP, MRR, Precision@1 and Accuracy.

C Experiments and Results

C.1 Ablation: MLM-only Pre-training

Table 5 presents a more detailed comparison between models continuously pre-trained only with MLM and models using also the sentence-level classification loss functions we proposed in this paper.

D Qualitative Examples from AS2

We present some qualitative examples from the three public AS2 datasets. We highlight cases in which the baseline RoBERTa-Base model is unable to rank the correct answer in the top position, but where our model pretrained with SP is successful. The examples are provided in Table 6.

Model+ Data Sampling	ASNQ			WikiQA			TREC-QA			WQA		
	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR
RoBERTa-Base	61.8 (0.2)	66.9 (0.1)	73.1 (0.1)	78.3 (2.8)	85.8 (1.3)	87.2 (1.3)	90.0 (1.9)	89.7 (0.7)	94.4 (1.1)		Baseline	
+ SSP Data (**MLM-only**)	63.4 (0.4)	67.1 (0.2)	73.8 (0.2)	76.7 (0.9)	84.5 (0.7)	85.8 (0.7)	87.4 (1.3)	88.8 (0.6)	93.1 (1.0)	-0.6%	-0.2%	-0.3%
+ SSP	**64.1** (0.3)	**68.1** (0.2)	**74.5** (0.3)	**82.9** (0.7)	**88.7** (0.3)	**89.9** (0.4)	**88.5** (1.2)	**89.3** (0.7)	**93.6** (0.6)	**+0.2%**	**+0.6%**	**+0.3%**
+ SP Data (**MLM-only**)	62.8 (0.3)	67.2 (0.2)	73.7 (0.2)	76.8 (1.6)	84.7 (0.8)	86.2 (0.7)	88.8 (1.3)	89.8 (0.3)	93.7 (0.9)	-1.0%	-0.4%	-0.6%
+ SP	**64.1** (0.2)	**68.3** (0.1)	**74.5** (0.2)	**81.0** (0.8)	**87.7** (0.3)	**88.9** (0.4)	**90.9** (2.6)	**90.1** (0.8)	**94.7** (1.3)	**+0.4%**	**+0.7%**	**+0.5%**
+ PSD Data (**MLM-only**)	**64.1** (0.5)	67.3 (0.2)	73.7 (0.2)	79.1 (1.6)	85.6 (1.4)	87.1 (1.2)	87.1 (2.8)	89.6 (1.0)	92.7 (1.3)	-1.3%	-0.3%	-0.6%
+ PSD	62.6 (0.4)	**67.7** (0.2)	73.7 (0.3)	**80.5** (1.6)	**86.4** (1.1)	**88.0** (1.0)	**90.3** (1.3)	**90.3** (0.5)	**95.1** (0.7)	**+0.4%**	**+0.7%**	**+0.5%**

Table 5: Results (with std. dev. across 5 runs in parentheses) of our pretrained transformer models when fine-tuned on AS2 datasets with MLM-only pre-training. SSP, SP and PSD refer to our pretraining objectives. Results on WQA are relative to RoBERTa baseline. We highlight in bold and underline results like in Table 1.

ASNQ
Q: how many players in football hall of fame
A1: Two coaches (Marv Levy , Bud Grant) , one administrator (Jim Finks) , and five players (Warren Moon , Fred Biletnikoff , John Henry Johnson , Don Maynard , Arnie Weinmeister) who spent part of their careers in the Canadian Football League (CFL) have been inducted ; two of which have been inducted into the Canadian Football Hall of Fame : Warren Moon and Bud Grant.
A2: As of 2018 , 318 individuals have been elected .
A3: Six players or coaches who spent part of their careers in the short-lived United States Football League (USFL) have been inducted .
A4: Current rules of the committee stipulate that between four and eight individuals are selected each year .
A5: Fifteen inductees spent some of their playing career in the All - America Football Conference during the late 1940s .
WikiQA
Q: how are antibodies used in
A1: Antibodies are secreted by a type of white blood cell called a plasma cell .
A2: An antibody (Ab), also known as an immunoglobulin (Ig), is a large Y-shaped protein produced by B-cells that is used by the immune system to identify and neutralize foreign objects such as bacteria and viruses .
A3: Using this binding mechanism, an antibody can tag a microbe or an infected cell for attack by other parts of the immune system, or can neutralize its target directly (for example, by blocking a part of a microbe that is essential for its invasion and survival).
A4: Antibodies can occur in two physical forms, a soluble form that is secreted from the cell, and a membrane -bound form that is attached to the surface of a B cell and is referred to as the B cell receptor (BCR).
A5: The BCR is only found on the surface of B cells and facilitates the activation of these cells and their subsequent differentiation into either antibody factories called plasma cells , or memory B cells that will survive in the body and remember that same antigen so the B cells can respond faster upon future exposure.
TREC-QA
Q: Where is the group Wiggles from ?
A1: Let 's now give a welcome to the Wiggles , a goofy new import from Australia .
A2: The Wiggles are four effervescent performers from the Sydney area : Anthony Field , Murray Cook , Jeff Fatt and Greg Page .
A3: In Australia , the Wiggles is like really huge .
A4: His group had kids howling with joy with routines involving Dorothy the Dinosaur , Henry the Octopus and Wags the Dog .
A5: While relatively new to the American scene , the Wiggles seem to be on to something , judging by kids ' reactions to the group 's belly-slapping shows .

Table 6: Qualitative examples from AS2 datasets where the baseline RoBERTa-Base model is unable to rank a correct answer for the question at the top position, but our SP pre-trained model can (top ranked correct answer by SP). Here we present the top ranked answers $\{A1, \ldots, A5\}$ in the order given by the RoBERTa-Base model. For all these examples we highlight the top ranked answer by the baseline RoBERTa-Base model in red since it is incorrect, and any other correct answer in green.