Context-Aware Transformer Pre-Training for Answer Sentence Selection

Anonymous ACL submission

Abstract

Answer Sentence Selection (AS2) is one of the main components for building an accurate Question Answering pipeline. AS2 models rank a set of candidate sentences based on how likely they answer a given question. The state of the art in AS2 exploits pre-trained transformers by transferring them on large annotated datasets, while using local contextual information around the candidate sentence. In this paper, we propose three pre-training objectives designed to mimic the downstream fine-tuning task of contextual AS2. This allows for specializing language models when fine-tuning for contextual AS2. Our experiments with continuous pre-training of RoBERTa and ELECTRA using two public and two large-scale industrial datasets show that our pre-training approaches can improve the accuracy of baseline of contextual AS2 by up to 2.4%.

1 Introduction

Answer Sentence Selection (AS2) is a fundamental task in QA, which consists of re-ranking a set of answer sentence candidates according to how likely they correctly answer a given question. From a practical standpoint, AS2-based QA systems can operate under much lower latency constraints than corresponding Machine Reading (MR) based QA systems. This is because AS2 systems process several sentences/documents in parallel, while MR systems parse the entire document/passage in a sliding window fashion before finding the answer (Garg and Moschitti, 2021).

Modern AS2 systems (Garg et al., 2020; Laskar et al., 2020) use transformers to cross-encode question and answer candidates together. Recently, Lauriola and Moschitti (2021) have demonstrated that performing answer ranking using only the candidate sentence is sub-optimal, for e.g., the answer sentence may contain unresolved coreference with entities, or the sentence may lack specific context for answering the question. Several works (Ghosh et al., 2016; Tan et al., 2018; Han et al., 2021) have explored performing AS2 using context around answer candidates (for example, adjacent sentences) towards improving performance. Local contextual information, i.e., the previous and next sentences of the answer candidates, can help coreference disambiguation, and provide additional knowledge to the model. This helps to rank the best answer at the top, with minimal increase in compute requirements.

Some research works (Lauriola and Moschitti, 2021; Han et al., 2021) have directly used existing pre-trained models such BERT, RoBERTa, etc. for contextual AS2, by fine-tuning them on a input text constituted by several sentences with different roles, i.e., the question, answer candidate, and local context (previous and following sentences around the candidate). This structured input creates practical challenges during fine-tuning, as standard pre-training approaches do not align well with the downstream contextual AS2 task, e.g., the language model does not know the role of each of the multiple sentences in the input. In other words, the extended sentence-level embeddings have to be learnt directly during fine-tuning, causing underperformance empirically.

In this paper, we tackle the aforementioned issues by designing three pre-training objectives that structurally align with the final contextual AS2 task, and can help improve the performance of language models when fine-tuned for AS2. Our pre-training objectives exploit information in the structure of paragraphs and documents to pre-train the context slots in the transformer text input. We evaluate our strategies on two popular pre-trained transformers using two large public and two large industrial datasets. The results show that our approaches can effectively adapt transformers to process contextualized input, thus improving model accuracy. Compared to baselines, our structural pre-training improves the models' accuracy by up to 2.4%. We plan to release code and pre-trained models.

2 Related Work

Answer Sentence Selection: TANDA (Garg et al., 2020) established the state of the art for AS2 using a large dataset (ASNQ) for transfer learning. Other approaches for AS2 include: (Bonadiman and Moschitti, 2020), that trains separate encoders for question and answer candidates, and (Yoon et al., 2019), which uses compare-aggregate and clustering to gather more information for each candidate.

Contextual AS2: Ghosh et al. (2016) refine LSTMs to accept answer candidates and topics in input, allowing higher accuracy in tasks such as next sentence selection. Tan et al. (2018) use GRU networks to model answer candidates and local context, improving performance on two AS2 datasets. The first contextualized transformer for AS2 was proposed by Lauriola and Moschitti (2021), which use both local and global document-level context to better disambiguate between answer candidates. Han et al. (2021) use unsupervised similarity matching techniques to extract relevant context for answer candidates from the document to enhance ranking capabilities of their models.

Pre-training Objectives: Sentence-level objectives such as Next Sentence Prediction (Devlin et al., 2019), Sentence Order Prediction (Lan et al., 2020) and Span Prediction (Joshi et al., 2019) have been widely explored for transformers (along with token-level objectives (Devlin et al., 2019; Liu et al., 2019)) to improve accuracy for downstream sequence classification tasks. However, the majority of these objectives are agnostic of the downstream tasks. End task-aware pre-training has been studied for summarization (Rothe et al., 2021), dialogue systems (Li et al., 2020), passage retrieval (Gao and Callan, 2021) and multi-task learning (Dery et al., 2021). Lee et al. (2019), Chang et al. (2020) and Sachan et al. (2021) use the Inverse Cloze task to improve retrieval performance for bi-encoders, by exploiting paragraph structure via self-supervised objectives. For AS2, recently Di Liello et al. (2022a) proposed paragraph-aware pre-training for joint classification of multiple candidates. Di Liello et al. (2022b) propose a sentence-level pre-training paradigm for AS2 by exploiting document and paragraph structure. However, these works do not consider the structure of the downstream task (specifically contextual AS2). To the best of our knowledge, ours is the first work to study transformer pre-training strategies for AS2 augmented with context using cross-encoders.

3 Contextual AS2

AS2: Given a question q and a set of answer candidates $S = \{s_1, \ldots, s_n\}$, the goal is to find the best s_k that answers q. This is typically done by learning a binary classifier $\mathbf{C}$ of answer correctness by independently feeding the pairs $(q, s_i), i \in \{1, \ldots, n\}$ as input to $\mathbf{C}$, and making $\mathbf{C}$ predict whether s_i correctly answers q or not. At inference time, we find the best answer for q by selecting the answer candidate s_k which scores the highest probability of correctness $k = \arg\max_i \mathbf{C}(q, s_i)$.
Contextual AS2: Contextual models for AS2 exploit additional context around answer candidates to improve the final accuracy. This has been shown to be effective (Lauriola and Moschitti, 2021) in terms of overcoming coreference disambiguation and lack of enough information to rank the best answer at the top. Different from the above case, contextual AS2 models receive as input a tuple (q, s_i, c_i) where c_i is the additional context corresponding to sentence candidate s_i. A popular option for c_i is to consider the sentences immediately before and after the answer candidate.

4 Context-aware Pre-training Objectives

We design a transformer pre-training task that aligns well with fine-tuning contextual AS2 models, both structurally and semantically. We exploit the division of large corpora in documents and the subdivision of documents in paragraphs as a source of supervision. We provide triplets of text spans (a, b, c) as model inputs when pre-training, which emulates the structure of (q, s_i, c_i) for contextual AS2 models, where a, b and c play the analogous role of the question, the candidate sentence (that needs to be classified), and the context (which helps in predicting (a, b) correctness), respectively. Formally, given a document D from the pre-training corpus, the task is to infer if a and b are two sentences extracted from the same paragraph $P \in D$. We called this objective: “Sentences in Same Paragraph (SSP)”. We design three different ways of choosing the appropriate contextual information c and then we present the details on how we automatically sample the text spans a, b and c from the pre-training documents.

Static Document-level Context (SDC) Here, we choose the context c to be the first paragraph P_0 of $D = \{P_0, .., P_n\}$ from which b is extracted. This is based on the intuition that the first paragraph acts as a summary of a document’s content

(Chang et al., 2020): this strong context can help the model at identifying if b is extracted from the same paragraph as a. We call this static document-level context since the contextual information c is constant for any b extracted from the same document D. Specifically, the positive examples are created by sampling a and b from a single random paragraph $P_i \in D, i > 0$. For the previously chosen a, we create hard negatives by randomly sampling a sentence b from different paragraphs $P_j \in D, j \neq i \wedge j > 0$. We set $c = P_0$ for this negative example as well since b still belongs to D. We create easy negatives for a chosen a by sampling b from a random paragraph P'_i in another document $D' \neq D$. In this case, c is chosen as the first paragraph P'_0 of D' since the context in the downstream AS2 task is associated with the answer candidate, and not with the question.

Dynamic Paragraph-level Context (DPC) We dynamically select the context c to be the paragraph from which the sentence b is extracted. We create positive examples by sampling a and b from a single random paragraph $P_i \in D$, and we set the context as the remaining sentences in P_i, i.e., $c = P_i \setminus \{a, b\}$. Note that leaving a and b in P_i would make the task trivial. For the previously chosen a, we create hard negatives by sampling b from another random paragraph $P_j \in D, j \neq i$, and setting $c = P_j \setminus \{b\}$. We create easy negatives for a chosen a by sampling b from a random P'_i in another document $D' \neq D$, and setting $c = P'_i \setminus \{b\}$.

Dynamic Sentence-level Local Context (DSLC) We choose c to be the local context around the sentence b, i.e, the concatenation of the previous and next sentence around b in $P \in D$. To deal with corner cases, we require at least one of the previous or next sentences of b to exist (e.g., the next sentence may not exist if b is the last sentence of the paragraph P). We term this DSLC as the contextual information c is specified at sentence-level and changes correspondingly to every sentence b extracted from D. We create positive pairs similar to SDC and DPC by sampling a and b from the same paragraph $P_i \in D$, with c being the local context around b in P_i (and $a \notin c$). We automatically discard paragraphs that are not long enough to ensure the creation of a positive example. We generate hard negatives by sampling b from another $P_j \in D, j \neq i$, while for easy negatives, we sample b from a $P'_i \in D', D' \neq D$ (in both cases c is set as the local context around b).

5 Datasets

Pre-Training To perform a fair comparison and avoid any improvement arising from the usage of additional pre-training data, we use the same pre-training corpus as RoBERTa (Liu et al., 2019). This includes the English Wikipedia, the BookCorpus (Zhu et al., 2015), OpenWebText (Gokaslan and Cohen, 2019) and CC-News[1]. We transform the datasets above to implement the pre-training objectives that we described in Section 4.

Contextual AS2 We evaluate our pre-trained models on two public and two industrial datasets for contextual AS2[2]. For all datasets, we use the standard 'clean' setting, by having at least one positive and one negative candidate per question in the dev. and test sets. We measure performance using P@1 (Precision-at-1), MAP and MRR. Dataset statistics are presented in Appendix A.2.

For testing our pre-training models on the downstream task, we used the following AS2 datasets:

- **ASNQ** is a large scale AS2 dataset (Garg et al., 2020) derived from NQ (Kwiatkowski et al., 2019). The questions are user queries from Google search, and answers are extracted from the top ranked Wikipedia page. We extract the contextual information for each answer candidate and use the data splits of Lauriola and Moschitti (2021).
- **NewsAS2** is a large scale AS2 dataset created from NewsQA (Trischler et al., 2017), an MR dataset, following the same procedure used by Garg et al. for ASNQ. The dataset contains ∼70K questions generated by humans and answer candidates extracted from the *CNN/Daily Mail* corpus.
- **IQAD** is a large scale industrial dataset containing de-identified questions asked by users to a popular commercial virtual assistant. IQAD contains ∼220k questions where answers are retrieved from a large web index (∼1B web pages) using Elasticsearch. We use two different evaluation benchmarks for IQAD: (i) **IQAD Bench 1**: Contains 2.2k questions with 15 answer candidates annotated for correctness by crowd workers, (ii) **IQAD Bench 2**: Contains 2k questions with 15 answer candidates annotated with explicit fact verification guidelines for correctness by crowd workers. (Our manual analysis indicates a higher annotation quality for QA pairs in Bench 2 than Bench 1). Results on

[1] The STORIES (Trinh and Le, 2018) dataset is no longer publicly available, and thus we ignore it

[2] We do not use popular AS2 datasets such as WikiQA (Yang et al., 2015), TREC-QA (Wang et al., 2007) due to contextual information not being available for them.

Model	Context used	ASNQ			NewsAS2			IQAD Bench 1			IQAD Bench 2		
		MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
ELECTRA-Base	✗	69.3 (0.0)	75.1 (0.1)	65.0 (0.2)	81.3 (0.2)	84.2 (0.1)	75.6 (0.2)	Baseline			Baseline		
ELECTRA-Base ♣	✓	72.3 (0.6)	77.9 (0.8)	68.1 (0.8)	82.0 (0.4)	84.6 (0.2)	76.0 (0.5)	-0.6%	-0.6%	-1.0%	-0.4%	-0.4%	-0.9%
(Ours) ELECTRA-Base + SSP (SDC)	✓	**74.7** (0.5)	**79.5** (0.3)	69.6 (0.3)	82.7 (0.2)	85.3 (0.3)	77.0 (0.4)	**+1.2%**	**+0.6%**	**+0.6%**	**+0.9%**	**+0.9%**	**+1.4%**
(Ours) ELECTRA-Base + SSP (DPC)	✓	74.4 (0.2)	**79.5** (0.2)	**70.5** (0.2)	82.7 (0.5)	**85.6** (0.4)	**77.3** (0.7)	+0.4%	-0.3%	-0.6%	+0.4%	+0.2%	+0.1%
(Ours) ELECTRA-Base + SSP (DSLC)	✓	74.3 (0.3)	79.4 (0.5)	70.0 (0.8)	**82.8** (0.4)	85.5 (0.4)	**77.3** (0.5)	+1.0%	+0.5%	+0.6%	+0.2%	+0.2%	0.0%
(Ours) ELECTRA-Base + SSP (All)	✓	73.8 (0.4)	78.7 (0.3)	68.8 (0.4)	82.7 (0.2)	85.4 (0.2)	77.2 (0.3)	+0.1%	-0.1%	-0.4%	+0.1%	+0.1%	-0.1%
RoBERTa-Base	✗	68.2 (0.5)	74.2 (0.3)	63.5 (0.5)	81.7 (0.1)	84.4 (0.0)	76.2 (0.2)	+0.6%	+0.5%	+0.1%	+0.7%	+0.9%	+1.3%
RoBERTa-Base ♣	✓	71.6 (0.6)	77.3 (0.5)	67.6 (0.6)	82.4 (0.2)	85.1 (0.4)	76.6 (0.7)	+0.4%	+0.4%	0.0%	+1.1%	+0.9%	+1.7%
(Ours) RoBERTa-Base + SSP (SDC)	✓	73.1 (0.5)	78.4 (0.6)	68.7 (0.8)	82.8 (0.1)	85.4 (0.2)	76.9 (0.2)	**+1.7%**	**+1.8%**	**+3.0%**	+1.0%	+0.9%	+1.7%
(Ours) RoBERTa-Base + SSP (DPC)	✓	**73.2** (0.4)	**78.5** (0.3)	**69.2** (0.5)	82.3 (0.1)	84.9 (0.1)	76.0 (0.1)	+0.4%	+0.6%	+1.2%	+1.2%	**+1.5%**	**+2.7%**
(Ours) RoBERTa-Base + SSP (DSLC)	✓	72.9 (0.4)	78.4 (0.2)	69.0 (0.3)	82.6 (0.2)	85.3 (0.1)	77.0 (0.2)	+0.6%	+0.9%	+1.5%	+1.0%	+0.9%	+1.4%
(Ours) RoBERTa-Base + SSP (All)	✓	72.9 (0.6)	77.9 (0.6)	68.2 (0.8)	**83.0** (0.2)	**85.6** (0.4)	**77.3** (0.5)	+1.2%	+1.5%	+2.4%	**+1.4%**	+1.3%	+2.2%

Table 1: Results (std. dev. in parenthesis) on AS2. Models with ♣ are from (Lauriola and Moschitti, 2021). ✓ and ✗ denote whether local contextual information was used in fine-tuning. SDC, DPC and DSLC indicate the pre-training variants of the SSP task that we propose. Best results are in bold while we underline statistically significant improvements over the two contextual baselines (♣) using a Student *t*-test with 95% of confidence level.

IQAD are presented relative to a baseline due to the data being internal.

6 Experiments

Continuous Pre-Training We use RoBERTa-Base and ELECTRA-Base[3] public checkpoints, and perform continuous pre-training using our objectives for ∼10% of the compute used by the original models. Extensive details are given in Appendix C. We experiment with each of our pre-training objectives independently, as well as combining all of them with the two model architectures.

Fine-Tuning We fine-tune each continuously pre-trained model on all the AS2 datasets. As baselines, we consider (i) standard pairwise-finetuned AS2 models, using only the question and the answer candidate, and (ii) contextual fine-tuned AS2 models from (Lauriola and Moschitti, 2021), that use the question, answer candidate and local context.

Results Table 1 summarizes the results of our experiments averaged across 5 runs. On ASNQ, our pre-trained models get 3.8-5.5% improvement in P@1 over the baseline using only the question and answer. Our models also outperform the stronger contextual AS2 baselines (1.6% with RoBERTa and 2.4% with ELECTRA), indicating that our task-aware pre-training can help improve the downstream fine-tuning performance. On NewsAS2, we observe a similar trend in results, where all our models (except one) outperform both the pair-wise and contextual baselines.

On IQAD, we observe that the contextual baseline performs at-par or lower than the non-contextual baseline, indicating that off-the-shelf transformers cannot effectively exploit the context available for this dataset. The answer candidates and context for IQAD are extracted from millions of web documents. Thus, learning from the context in IQAD is a harder task than learning from it on ASNQ, where the context belongs to a single Wikipedia document. Our pre-trained models help to process the diverse and possibly noisy context of IQAD, and produce a significant improvement in P@1 over the contextual baseline.

The DPC and DSLC approaches align well (often having overlapping or identical contexts for the same (a, b) input), explaining their comparable performance across all datasets. In SDC, the context c can potentially be very different from (a, b), and this may help in exploiting information from multiple documents/domains as is the case for IQAD. For these reasons, we believe DPC and DSLC should be used when answer candidates are extracted from withing the same document, while SDC works best with candidates collected across multiple documents. We present an extended discussion of our results in Appendix E. Also, we observe that combining all the objectives together does not always outperform the individual objectives, which is probably due to the different pre-training ways of sampling context being heterogeneous from each other and not aligning well.

7 Conclusions

In this paper, we have proposed three pre-training strategies for transformers, which (i) are aware of the downstream task of contextual AS2, and (ii) use the document and paragraph structure information to define effective objectives. Our experiments on two public and two industrial datasets using two transformer models show that our pre-training strategies can provide significant improvement over the contextual AS2 models.

[3] Due to compute limitations, we don't extend our pre-training experiments to Large models (each pre-training run will take ∼10 days on 8 A100-GPUs each with 40GB RAM).

References

Daniele Bonadiman and Alessandro Moschitti. 2020. A study on efficiency, accuracy and document structure for answer sentence selection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5211–5222, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval.

Lucio M. Dery, Paul Michel, Ameet Talwalkar, and Graham Neubig. 2021. Should we be pre-training? an argument for end-task aware training as an alternative.

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. 2022a. 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*, Seattle, Washington. Association for Computational Linguistics.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022b. Pre-training transformer models with sentence-level objectives for answer sentence selection.

William Falcon et al. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3(6).

Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval.

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.

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks.

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

Rujun Han, Luca Soldaini, and Alessandro Moschitti. 2021. Modeling context in answer sentence selection systems on a latency budget.

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

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. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

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. In *ECIR 2021*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2020. Task-specific objectives of pretrained language models for dialogue adaptation.

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.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

Sascha Rothe, Joshua Maynez, and Shashi Narayan. 2021. A thorough evaluation of task-specific pretraining for summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 140–145, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering.

Chuanqi Tan, Furu Wei, Qingyu Zhou, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. Context-aware answer sentence selection with hierarchical gated recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):540–549.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *ArXiv*, abs/1806.02847.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

Appendix

A Datasets

A.1 Pre-training

We preprocess Wikipedia, the BookCorpus, CC-News and OpenWebText by filtering away: (i) sentences having a length smaller than 20 characters, (ii) paragraphs shorter than 60 characters and (iii) documents shorter than 200 characters. We split paragraphs in sequences of sentences using the NLTK tokenizer (Loper and Bird, 2002) and we create the datasets for continuous pre-training following the definitions in Section 4.

For each objective, we sample randomly up to 2 hard negatives and additional easier negatives until the total number is 4. Instead of reasoning in terms of sentences, we designed our objectives to create a and b as small spans composed of 1 or more contiguous sentences. For a, we keep the length equal to 1 sentence because it emulates the question, which usually is just a single sentence. For b, we randomly sample the length between 1 and 3. The length of the context c cannot be decided a priori because it depends on the specific pre-training objective and the length of the paragraph.

All the resulting continuous pre-training datasets are about 300GB in size (uncompressed) and contain around 350M training examples each.

A.2 Fine-Tuning

The statistics on the number of unique questions and question-answer pairs for each fine-tuning dataset are provided in Table 2. While ASNQ has a huge number of negatives for each question (more than 300 on average), NewsAS2 has a smaller number of answer candidates per question (25 on average). Note that we do not use the contextual WikiQA dataset for our experiments due to missing contextual information (on contacting the authors of (Lauriola and Moschitti, 2021), we found that this dataset is not available anymore).

NewsAS2 was created by splitting each document in NewsQA into individual sentences with the NLTK tokenizer (Loper and Bird, 2002). Then, for each sentence, we assigned a positive label if it contained at least one of the annotated answers for that document, a negative label otherwise. This lead to a datasets with 1.69% positives sentences per query in the training set, 1.66% in the dev set and 1.68% in the test set. We will release this NewsAS2 dataset along with code and models from our paper.

Dataset	Train		Dev		Test	
	#Q	#QA	#Q	#QA	#Q	#QA
ASNQ	57242	20377568	1336	463914	1336	466148
IQAD	221334	3894129	2434	43369	2252 2088	38587 33498
NewsAS2	71561	1840533	2102	51844	2083	51472

Table 2: Number or unique questions and question-answer pairs in the fine-tuning datasets. IQAD Bench 1 and Bench 2 sizes are mentioned in the Test set column corresponding to IQAD.

B Frameworks & Infrastructure

Our framework is based on (i) HuggingFace Transformers (Wolf et al., 2020) for model architecture, (ii) HuggingFace Datasets (Lhoest et al., 2021) for data processing, (iii) PyTorch-Lightning for distributed training (Falcon et al., 2019) and (iv) TorchMetrics for AS2 evaluation metrics (Detlefsen et al., 2022).

We performed our pre-training experiments for every model on 8 NVIDIA A100 GPUs with 40GB of memory each, using *fp16* for tensor core acceleration.

C Continuous Pre-Training

We experiment with RoBERTa-Base and ELECTRA-Base public checkpoints. RoBERTa-Base contains 124M parameters while ELECTRA-Base contains 33M parameters in the generator and 108M in the discriminator.

We do continuous pre-training starting from the aforementioned models for 400K steps with a batch size of 4096 examples and a triangular learning rate with a peak value of 10^{-4} and 10K steps of warm-up. In order to save resources, we found it beneficial to reduce the maximum sequence length to 128 tokens. In this setting, our models see about 210B additional tokens each, which are exactly the 10% of those used in the original RoBERTa pre-training. Moreover, in terms of complexity our objectives are more efficient because the attention computational complexity grows quadratically in the sequence length, which in our case is 4 times smaller.

We use cross-entropy as the loss function for all our pre-training and fine-tuning experiments. Specifically, for RoBERTa pre-training we sum the MLM and our proposed binary classification losses with equal weights (1.0). For ELECTRA pre-training, we sum three losses: MLM loss with a weight of 1.0, the Token Detection loss with a weight of 50.0, and our proposed binary classification losses with a weight of 1.0.

Model	Hyper-parameter	ASNQ	NewsAS2	IQAD
RoBERTa	Batch size	2048	256	256
	Peak LR	1e-05	5e-06	1e-05
	Warmup steps	10K	5K	5K
	Epochs	6	8	10
ELECTRA	Batch size	1024	128	256
	Peak LR	1e-05	1e-05	2e-05
	Warmup steps	10K	5K	5K
	Epochs	6	8	10

Table 3: Hyper-parameters used to fine-tune RoBERTa and ELECTRA on the AS2 datasets. The best hyper-parameters has been chosen based on the MAP results on the validation set.

During continuous pre-training, we feed the text tuples (a, b, c) (as described in Section 4) as input to the model in the following format: '[CLS]a[SEP]b[SEP]c[SEP]'.

To provide independent sentence/segment ids to each of the inputs a, b and c, we initialize the sentence embeddings layers of RoBERTa and ELECTRA from scratch, and extend them to an input size of 3.

The pre-training of every model obtained by combining ELECTRA and RoBERTa architectures with our contextual pre-training objectives took around 3.5 days each on the machine configuration described in Appendix B. All the dataset preparation required 10 hours over 64 CPU cores.

D Fine-Tuning

The most common paradigm for AS2 fine-tuning is to consider publicly available pre-trained transformer checkpoints (pre-trained on large amounts of raw data) and fine-tune them on the AS2 datasets. Using our proposed pre-training objectives, we are proposing stronger model checkpoints [4] which can improve over the standard public checkpoints, and can be used as the initialization for downstream fine-tuning for contextual AS2.

To fine-tune our models on the downstream AS2 datasets, we found it is beneficial to use a very large batch size for ASNQ and a smaller one for IQAD and NewsAS2. Moreover, for every experiment we used a triangular learning rate scheduler and we did early stopping on the development set if the MAP did not improve for 5 times in a row. We fixed the maximum sequence length to 256 tokens in every run, and we repeated them 3 times with different initial random seeds. We did not use weight decay but we clipped gradients larger than 1.0 in absolute value. More specifically, for the learning rate we tried all values in $\{5 * 10^{-6}, 10^{-5}, 2 * 10^{-5}\}$ for RoBERTa and in $\{10^{-5}, 2 * 10^{-5}, 5 * 10^{-5}\}$ for ELECTRA. Regarding the batch size, we tried values $\{512, 1024, 2048, 4096\}$ for ASNQ and $\{46, 128, 256, 512\}$ for IQAD and NewsAS2. More details about final hyper-parameter are given in Table 3.

For the pair-wise models, we format inputs as '[CLS]q[SEP]s_i[SEP]', while for contextual models we build inputs of the form '[CLS]q[SEP]s_i[SEP]c_i[SEP]'.

We do not use extended sentence/segment ids for the non-contextual baselines and retain the original model design: (i) disabled segment ids for RoBERTa and (ii) only using 2 different sentence/segment ids for ELECTRA. For the fine-tuning of our continuously pre-trained models as well as the contextual baseline, we use three different sentence ids corresponding to q, s and c for both RoBERTa and ELECTRA.

Finally, differently from pre-training, in fine-tuning we always provide the previous and the next sentence as context for a given candidate.

The contextual fine-tuning of every models on ASNQ required 6 hours per run on the machine configuration described in Appendix B. For the other fine-tuning datasets, we use a single GPU per experiment, which took less than 2 hours.

E Additional Discussion of Results

Here we explain the difference in performance we observe from our three pre-training objectives on different AS2 datasets. The AS2 datasets we consider for our experiments have significantly different structures: specifically, ASNQ and NewsAS2 have answer candidates being extracted from a single document (Wikipedia and CNN Daily Mail article respectively), while IQAD has answer candidates being extracted from multiple documents. This also results in the context for the former being more homogeneous (context for all candidates for a question is extracted from the same document), while for the latter the context is more heterogeneous (extracted from multiple documents for different answer candidates).

Our DPC and DSLC pre-training approaches are well aligned in terms of the context that is used to help the SSP predictions. The former uses the remainder of the paragraph P as context (after removing a and b), while the latter uses the sentence previous and next to b in P. We observe empiri-

[4] We plan to release our code and pre-trained model checkpoints after the anonymity period.

cally that the contexts for DPC and DSLC often overlap partially, and are sometimes even identical (considering average length of paragraphs in the pre-training corpora is 4 sentences). This explains why models pre-trained using both these approaches perform comparably in Table 1 (with only a very small gap in P@1 performance).

On IQAD, we observe that the SDC approach of providing context for SSP outperforms the DPC and DSLC approaches for pre-training. In SDC, the context c can potentially be very different from a and b (as it corresponds to the first paragraph of the document), and this can aid exploiting information and effectively ranking answer candidates from multiple documents (possibly from different domains) like for IQAD.

F Qualitative Examples

In Table 4 we show a comparison of the ranking produced by our models and that by the contextual baselines on some questions selected from the ASNQ test set.

ELECTRA	
Q	**how many games does a team have to win for the world series**
A₁	Seven games were played, with the Astros victorious after game seven, played in Los Angeles.
A₂	In 1985, the format changed to best-of-seven.
A₃	Since then, the 2011, 2014, and 2016 World Series have gone the full seven games.
A₄	The winner of the World Series championship is determined through a best-of-seven playoff, and the winning team is awarded the Commissioner's Trophy.
A₅	The Houston Astros won the 2017 World Series in 7 games against the Los Angeles Dodgers on November 1st, 2017, winning their first World Series since their creation in 1962.
RoBERTa	
Q	**where are trigger points located in the body**
A₁	Myofascial pain is associated with muscle tenderness that arises from trigger points, focal points of tenderness, a few millimeters in diameter, found at multiple sites in a muscle and the fascia of muscle tissue.
A₂	Myofascial trigger points, also known as trigger points, are described as hyperirritable spots in the fascia surrounding skeletal muscle.
A₃	Trigger points form only in muscles.
A₄	These in turn can pull on tendons and ligaments associated with the muscle and can cause pain deep within a joint where there are no muscles.
A₅	They form as a local contraction in a small number of muscle fibers in a larger muscle or muscle bundle.

Table 4: Some qualitative examples from ASNQ test set where our ELECTRA and RoBERTa models with DSLC contextual continuous pre-training were able to rank the correct candidate in the top position while the contextual baselines failed. The answer candidates are shown ranked by the ordering produced by the contextual baselines. Other positive candidates answers are colored in light green.

G Discussion of Limitations

Our proposed pre-training approaches require access to large GPU resources (pre-training is performed on 350M training samples for large language models containing 100's of millions of parameters). Additionally, the pre-training takes a long time duration to finish (several days even on a large number of NVIDIA A100 GPUs), which highlights that this procedure cannot easily be re-done with newer data being made available in an online setting. However the benefit of our approach is that once the pre-training is complete, our released model checkpoints can be directly fine-tuned (even on smaller target datasets) for the downstream contextual AS2 task. For the experiments in this paper, we only consider datasets from the English language, however we conjecture that our techniques should work similarly for languages with limited morphology, like English. Finally, we believe the three proposed objectives could be better combined in a multi-task training scenario where the model has to jointly predict the task and the label. At the moment, we left this as a future research direction.