Answer Generation for Retrieval-based Question Answering Systems

Chao-Chun Hsu[1]*, **Eric Lind**[2], **Luca Soldaini**[2], **Alessandro Moschitti**[2]
[1]University of Chicago, [2]Amazon Alexa
chaochunh@uchicago.edu, {lssoldai,ericlind,amosch}@amazon.com

Abstract

Recent advancements in transformer-based models have greatly improved the ability of Question Answering (QA) systems to provide correct answers; in particular, answer sentence selection (AS2) models, core components of retrieval-based systems, have achieved impressive results. While generally effective, these models fail to provide a satisfying answer when all retrieved candidates are of poor quality, even if they contain correct information. In AS2, models are trained to select the best answer sentence among a set of candidates retrieved for a given question. In this work, we propose to generate answers from a set of AS2 top candidates. Rather than selecting the best candidate, we train a sequence to sequence transformer model to generate an answer from a candidate set. Our tests on three English AS2 datasets show improvement up to 32 absolute points in accuracy over the state of the art.

1 Introduction

Question answering systems are a core component of many commercial applications, ranging from task-based dialog systems to general purpose virtual assistants, e.g., Google Home, Amazon Alexa, and Siri. Among the many approaches for QA, AS2 has attracted significant attention in the last few years (Tymoshenko and Moschitti, 2018; Tian et al., 2020; Garg et al., 2020; Han et al., 2021). Under this framework, for a given question, a retrieval system is first used to obtain and rank a set of supporting passages; then, an AS2 model is used to estimate the likelihood of each sentence extracted from passages to be a correct answer, returning the one with the highest probability. This approach is favored in virtual assistant systems because full sentences are more likely to include the right context and sound natural, both of which are characteristics users value (Berdasco et al., 2019).

Q:	How a **water pump** works?
$\mathbf{c_1}$:	A small, electrically powered **pump**.
$\mathbf{c_2}$:	A large, electrically driven **pump** (electropump) for **waterworks** near the Hengsteysee, Germany.
$\mathbf{c_3}$:	A **pump is a device** that **moves fluids** (liquids or gases), or sometimes slurries, by **mechanical action**.
$\mathbf{c_4}$:	**Pumps** can be classified into three major groups according to the method they use to **move the fluid**: direct lift, displacement, and gravity **pumps**.
$\mathbf{c_5}$:	**Pumps** operate by some mechanism (typically reciprocating or rotary), and consume energy to perform **mechanical work** by **moving the fluid**.
G:	A **water pump is a device** that **moves fluids** by **mechanical action**.

Table 1: An example of a question Q and five answer candidates $c_1, \ldots, c_5$ from WikiQA (Yang et al., 2015) ranked by an AS2 system. Answer G generated by our best system is significantly more natural and concise than any extracted candidates.

AS2 models have shown great performance on academic benchmarks. However, these datasets fail to consider many essential qualities of a QA system which interacts directly with users, such as a virtual assistant. In some cases, extracted answer sentences contain the correct information, but the focus of the answer doesn't match the question; in others, the answer requires reasoning or contextual knowledge from the user or is very long and contains extraneous information. For example, in WikiQA (Yang et al., 2015), a widely used AS2 dataset, the answer *"Wind power is the conversion of wind energy into a useful form of energy, such as using wind turbines to make electrical power, windmills for mechanical power, wind pumps for water pumping... "* is considered a good answer for *"What can be powered by wind?"*, even though its formulation is burdensome to a user.

In this work, we explore a fundamentally differ-

* This work was completed while the author was an intern at Amazon Alexa.

ent approach to AS2. Rather than *selecting* the best candidate, we propose using a model to *generate* a suitable response for a user question. In so doing, we extend the traditional AS2 pipeline with a final generation stage that can recover correct and satisfying answers in cases where a ranking AS2 model fails to place an acceptable candidate at the top position or where a top candidate with the desired information is not a natural-sounding response to the query. Table 1 shows an example of our system: given the question, Q, and a list of candidates, $C_k = \{c_1, \ldots, c_5\}$ sorted by a state-of-the-art AS2 system, we use a sequence-to-sequence model to produce an answer G given Q and C_k as input. This approach, which we refer to as GenQA, addresses the limitations of AS2 systems by composing concise answers which may contain information from multiple sources.

Recent works have shown that large, transformer-based conditional generative models can be used to significantly improve parsing (Chen et al., 2020; Rongali et al., 2020), retrieval (De Cao et al., 2020; Pradeep et al., 2021), and classification tasks (Raffel et al., 2019). Our approach builds on top of this line of work by designing and testing generative models for AS2-based QA systems. In recent years, the use of generative approaches has been evaluated for other QA tasks, such as machine reading (MR) (Izacard and Grave, 2021; Lewis et al., 2020b) and question-based summarization (QS) (Iida et al., 2019; Goodwin et al., 2020; Deng et al., 2020). However, while related, these efforts are fundamentally different from the experimental setting described in this paper. Given a question, generative MR models are used to extract a short span (1-5 tokens) from a passage that could be used to construct an answer to a question. In contrast, AS2 returns a complete sentence that could be directly returned to a user.

QS systems are designed to create a general summary given a question and one or more related documents. Unlike QS, AS2-based QA systems need to provide specific answers; thus, the presence of even a small amount of unrelated information in a response could cause the answer sentence to be unsuitable. In contrast, we show that our approach can succinctly generate the correct information from a set of highly relevant sentence candidates.

In summary, our contribution is four-fold: (*i*) we introduce a new approach for AS2-based QA systems, which generates, rather than selects, an answer sentence; (*ii*) we illustrate how to adapt state-of-the-art models such as T5 (Raffel et al., 2019) and BART (Lewis et al., 2020a) for answer generation; (*iii*) we show[1] that our GenQA system improves over the state-of-the-art AS2-based systems by up to 32 accuracy points, as evaluated by human annotators; finally, (*iv*) we briefly explain why traditional generation metrics are not suited for evaluating AS2-based systems.

2 Datasets

We use four English datasets in our work, one related to generative QA and three to AS2. For a fair comparison between selector and generation methods, we re-evaluate the top answers returned by all models using a fixed set of annotators. All annotations were completed by company associates who are not part of our research group and had no knowledge of the systems. Annotators were required to mark an answer as correct if it was: (*i*) factually correct; (*ii*) natural-sounding; and (*iii*) required no additional information to be understood. All QA pairs were single annotated, as we determined sufficient agreement for this task in previous campaigns.

WikiQA by Yang et al. (2015) contains queries from Bing search logs and candidate answer sentences extracted from a relevant Wikipedia page. For evaluation, we used the dev. and test sets, which contain 126 and 243 unique questions and we re-annotated all of the resulting 569 QA pairs.[2]

Answer Sentence Natural Questions (ASNQ) introduced by Garg et al. (2020) was derived from the NQ dataset (Kwiatkowski et al., 2019) and consists of the questions which have a short answer span within a single sentence in a long answer span. The sentences containing the short answer are marked as correct and the other sentences in the document are marked as incorrect. We use the dev. and test splits introduced by Soldaini and Moschitti (2020) which contain 1,336 questions each. We re-annotated a total of 5,344 QA pairs.

WQA is an internal AS2 dataset created from a non-representative sample of questions asked by

[1] Our models, source code, and annotated data are available at: https://github.com/alexa/wqa-cascade-transformers.

[2] Due to time and annotation constraints, we were only able to annotate results for 100 queries from each of the dev. and test sets for our UQAT5 model

users of a virtual personal assistant in 2019[3]. For each question, we retrieved 500 pages from an index containing over 100M web documents. We then ranked candidate answers using a state-of-the-art AS2 system, and annotated up to 100 of them. In total, the training and dev. sets contain 3,074 queries and 189k QA pairs, while the test set contains 808 queries. For this effort, we re-annotated 4,847 QA pairs from the test set.

MS MARCO QA NLG (MSNLG) by Nguyen et al. (2016) is a subset of the MS MARCO dataset focused on generating natural language answers to user queries from web search result passages. It consists of 182k queries from Bing search logs, the ten most relevant passages retrieved for each query, and a well-formed answer synthesized by an annotator. This dataset is not designed for AS2, but it represents a large resource of succinct and clear answers, thus making it close to our AS2 task.

3 Generative QA Model (GenQA)

The AS2 task is defined as follows: Let q be an element of the question set, Q, and $C_q = \{c_1, \ldots, c_n\}$ be a set of candidates for q, e.g., sentences retrieved by a search engine, where $c_i \in C$, and C is a set of candidates. We model a selector $\mathcal{S} : Q \times C^n \rightarrow C$, such that $\mathcal{S}(q, C_q) = \text{argmax}_i\,(p(q, c_i))$, where $p(q, c_i)$ is the probability that c_i is a good answer. We also define $\mathcal{S}_k : Q \times C^n \rightarrow C^k$, such that, $\mathcal{S}_k$ selects the top k answers in descending order of $p(q, c_i)$.

State of the Art Throughout our experiments, we use TANDA (Garg et al., 2020) as our state-of-the-art selector $\mathcal{S}$. This AS2 model was trained as a binary classifier on (q, c_i) pairs using a sequential fine-tuning approach starting with ASNQ and finishing on a target dataset, e.g., WikiQA. Specifically, we use their pretrained RoBERTa Large model (Liu et al., 2019), as it achieved the best results on all datasets it was tested on.

3.1 Our Generative Approach

Instead of selecting the best candidate, we generate a new answer using the information from the top k answer candidates. Thus, our model is a function $\mathcal{G} : Q \times C^k \rightarrow G$, where G is the text that can be generated by the generator $\mathcal{G}$ from the question, any fragment of the retrieval set, the model's vocabulary, and knowledge stored in the model's parameters. Formally:

$$\mathcal{G}(q, C_q) = \mathcal{G}(q, C_k) = \mathcal{G}(q, \mathcal{S}_k(k, C_q)). \quad (1)$$

The example in Table 1 shows that we can generate a correct answer from a set of candidates which, as a whole, contain enough information to formulate a correct answer. We propose that a valid answer can be built by composing the *most promising* constituents coming from the different candidates in C_k. Intuitively, information repeated across multiple candidates is more *promising*; therefore, we hypothesize that a model trained on the same or similar generation task should be able to effectively exploit this form of repetition, even in cases where the same information is presented in a similar, but not identical manner. Further, recent works have shown that large transformer models hold a substantial amount of commonsense knowledge in their parameters (Roberts et al., 2020), which our model could leverage to perform inference across sentences in C_k, *e.g.*, associate *water* with *fluid* in the example in Table 1.

3.2 Fine-tuning GenQA

Given a pre-trained transformer seq2seq model, e.g., T5 (Raffel et al., 2019) or BART (Lewis et al., 2020a), we obtain $\mathcal{G}$ by fine-tuning on a large AS2 or QA generation dataset. For this purpose, we format our training data as a standard sequence-to-sequence/text-to-text task, where the source text is the question concatenated with the top five answer candidates, $(q, \mathcal{S}_{k=5})$, joined by newlines. When an answer composed by a human is available, such as in MSNLG, we use it as the output target. For cases where there is no composed answer, we randomly select a known-good candidate to be the target, remove it from the inputs and replace it with another candidate if one is available. We truncate the input text to 512 tokens and, at test time, we use beam search with a beam size of four and a maximum output length of 100 tokens.

4 Experiments

In this section, we first report on our experimental setup, then we show the results on fine-tuning GenQA, and finally, we report on the comparative results between AS2 and GenQA.

4.1 Setup

Models and Parameterization Our GenQA model is based on the T5 (Raffel et al., 2019) vari-

[3]The public version of WQA will be released in the short-term future. Please search for a publication with title *WQA: A Dataset for Web-based Question Answering Tasks* on arXiv.

ant of the UnifiedQA (UQAT5) model by Khashabi et al. (2020). We use the Large version of UQAT5, which has 770M parameters for all of our experiments. We compute training loss as the mean of the cross-entropy between the softmax probabilities over the output vocabulary and the one-hot encoded target answer. We fine-tune UQAT5 with a learning rate of $5E^{-5}$. We also experiment with the Large variant of BART (Lewis et al., 2020a), which is comprised of 400M parameters. This model was trained using same loss with learning rate $5E^{-6}$.

Evaluation We used accuracy as our primary metric for all our experiments and models. This is computed as the average number of questions a model answers correctly; for a selector $\mathcal{S}$, it is equivalent to Precision at 1. For $\mathcal{S}$, we also report Hit Rate at 5, which is the fraction of queries with at least one good candidate ranked five or less.

Beside human evaluation, we also experimented with automatic evaluation metrics such as BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) for GenQA. Such metrics have found little success in evaluating QA tasks (Chaganty et al., 2018; Chen et al., 2019), so we investigate whether that is the case for AS2 as well.

4.2 Results

How to Fine-tune GenQA? As described in Section 4.1, we tested two GenQA variants: one uses a UnifiedQA T5 (UQAT5) (Khashabi et al., 2020) as base model, while the other leverages BART-Large (Lewis et al., 2020a). Of the datasets used in this work, MSNLG and WQA are large enough for fine-tuning GenQA. Therefore, based on preliminary results, we tested four different strategies for training UQAT5: fine tuning on (*i*) WQA or (*ii*) MSNLG alone, (*iii*) combine the two datasets by alternating mini-batches during training, or (*iv*) follow the *transfer-then-adapt* strategy proposed by Garg et al. (2020): first fine-tune on MSNLG, then adapt to a AS2 using WQA.

Table 2 reports the results on the WQA test set, which are all relative to the performance of the state-of-the-art model (TANDA). First, we observe that all GenQA models reported in this table considerably outperform the best selector model, TANDA. This result shows that our generative approach can improve system based on AS2.

Comparing the accuracy of different training strategies applied to UQAT5, we achieve the best results when the model is trained on MSNLG alone

Model	Accuracy	BLEU	ROUGE-L
TANDA (Garg et al., 2020)	*baseline*	-	-
UQAT5 (AS2D)	+5.3%	40.8	55.7
UQAT5 (MSNLG)	**+19.9%**	20.2	39.7
UQAT5 (MSNLG+AS2D)	+13.6%	35.3	50.6
UQAT5 (MSNLG→AS2D)	+7.9%	40.6	54.8
BART-Large (MSNLG)	**+20.7%**	21.5	41.1

Table 2: Relative accuracy of different GenQA models and training configurations on the WQA dataset; both UQAT5 and BART perform best when finetuned on MSNLG only. As shown in previous work, automatic metrics (BLEU, ROUGE-L) do not correlate with human annotations (accuracy).

(+19.9% over TANDA baseline). While we were initially surprised by this result, as MSNLG is not designed for AS2, error analysis suggests that GenQA benefits from the high quality training data (concise answers written by annotators). Conversely, when training with WQA, we observed that GenQA tends to produce answers that, while correct, are not as natural-sounding. We plan to explore how to best leverage existing AS2 datasets for generative model training in future work. We also note that a GenQA BART-Large achieves comparable results to GenQA UQAT5 on WQA; in preliminary experiments, we found training strategies reported on UQAT5 to have similar effect on BART-Large.

When manually annotating results of our early tests, we found that BART was more likely to be extractive and copy input passages in their entirety while UQAT5 was more likely to compose new text and produce answers with textual overlap from multiple input candidates but was more likely to hallucinate content. We found that through hyperparameter tuning we could largely eliminate the hallucination from UQAT5 answers but we were unable to make BART more abstractive.

Similar to what has been observed in other QA tasks (Chaganty et al., 2018; Chen et al., 2019), we find that automatic metrics do not correlate with assessments from human annotators. This is due to the fact that neither BLEU nor ROUGE-L are designed to estimate whether an answer is clear and natural-sounding, instead rewarding candidates that have high overlap with reference answers. Most importantly, such overlap is a poor indicator of factual correctness.

Dataset	TANDA Acc.	TANDA Hit@5	TANDA Length	GenQA UQAT5 Acc.	GenQA UQAT5 Length
$\text{WikiQA}_{\text{DEV}}$	59.5	99.2	31.7 ± 13.7	**92.1**	14.9 ± 9.3
$\text{WikiQA}_{\text{TEST}}$	61.0	99.2	30.1 ± 12.4	**88.5**	14.6 ± 8.3
ASNQ_{DEV}	75.5	87.7	41.0 ± 122.4	**90.2**	13.9 ± 5.9
$\text{ASNQ}_{\text{TEST}}$	69.0	87.9	37.9 ± 51.5	**90.5**	13.9 ± 5.6

Table 3: Accuracy of our GenQA UQAT5 model compared to a state-of-the-art AS2 model by Garg et al. (2020). All answer candidates returned by the two models were re-annotated to ensure a fair comparison. Length is the average number of tokens in the answer.

Comparison between AS2 and GenQA Table 3 reports the results of TANDA and GenQA on two standard AS2 datasets, evaluated with manual annotation. We note that there is an impressive gap of over 20 absolute accuracy points on both development and test sets. This result is produced by two important properties of GenQA. First, it builds correct answers from a pool of correct and incorrect answers, and it can generate a good answer so long as the relevant information can be found anywhere in the top $k = 5$ candidates. This is a clear advantage over using TANDA alone, as Hit-Rate@5 of 99.2%, and 87.9% for WikiQA and ASNQ, respectively, ensures that GenQA often receives at least one correct answer as input.

Second, GenQA exhibits the ability to rewrite *unnatural* answers from a text snippet into an answer suitable for a conversation. For example, for the question *"What year did Isaac Newton die?"*, TANDA returns candidate *"Sir Isaac Newton (25 December 1642–20 March 1727) was an English physicist and mathematician"*. Although correct, no human would provide it in such a form. In contrast, GenQA composes a concise answer: *"Isaac Newton died in 1727"*.

Finally, Table 3 shows that the size of GenQA answers, in terms of words, is only 14 tokens, which is 2.7 times less than the 30-40 tokens from TANDA. This further suggests that GenQA can provide more concise and direct answers, which are preferable in a conversational context.

5 Conclusions

In this work we present GenQA, a generative approach for AS2-based QA systems. The main difference with recent MR-based generative systems is the capacity of our models to generate long answers. This comes from the use of AS2 candidates (complete sentences) as input to our generative approach. In contrast, MR systems, being mainly trained with short answers, e.g., noun phrases and named entities, mostly generate short answers.

We show that GenQA significantly outperforms state-of-the-art selector models for AS2 by up to 32 accuracy points by combining different pieces of information from the top k answer candidates. These results suggest promising directions for generative retrieval-based systems.

Acknowledgments

We thank Thuy Vu for setting up annotation procedures for the WQA dataset.

References

Ana Berdasco, Gustavo López, Ignacio Diaz, Luis Quesada, and Luis A. Guerrero. 2019. User experience comparison of intelligent personal assistants: Alexa, google assistant, siri and cortana. *Proceedings*, 31(1).

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-resource domain adaptation for compositional task-oriented semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5090–5100, Online. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7651–7658. AAAI Press.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: transfer and adapt pre-trained transformer models for answer sentence selection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7780–7788. AAAI Press.

Travis Goodwin, Max Savery, and Dina Demner-Fushman. 2020. Towards Zero-Shot Conditional Summarization with Adaptive Multi-Task Fine-Tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3215–3226, Online. Association for Computational Linguistics.

Rujun Han, Luca Soldaini, and Alessandro Moschitti. 2021. Modeling context in answer sentence selection systems on a latency budget. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3005–3010, Online. Association for Computational Linguistics.

Ryu Iida, Canasai Kruengkrai, Ryo Ishida, Kentaro Torisawa, Jong-Hoon Oh, and Julien Kloetzer. 2019. Exploiting background knowledge in compact answer generation for why-questions. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 142–151. AAAI Press.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1907.11692*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! A sequence to sequence architecture for task-oriented semantic parsing. In *WWW '20: The Web Conference*

2020, Taipei, Taiwan, April 20-24, 2020, pages 2962–2968. ACM / IW3C2.

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.

Zhixing Tian, Yuanzhe Zhang, Xinwei Feng, Wenbin Jiang, Yajuan Lyu, K. Liu, and Jun Zhao. 2020. Capturing sentence relations for answer sentence selection with multi-perspective graph encoding. In *AAAI*.

Kateryna Tymoshenko and Alessandro Moschitti. 2018. Cross-pair text representations for answer sentence selection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2162–2173, Brussels, Belgium. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.