Generating Answer Candidates for Quizzes and Answer-Aware Question Generators

Kristiyan Vachev
FMI, Sofia University
“St. Kliment Ohridski”
Sofia, Bulgaria
vachevkd@gmail.com

Momchil Hardalov
FMI, Sofia University
“St. Kliment Ohridski”
Sofia, Bulgaria
hardalov@fmi.uni-sofia.bg

Georgi Karadzhov
Department of Computer
Science and Technology
University of Cambridge, UK
gmk34@cam.ac.uk

Georgi Georgiev
Releva AI
Sofia, Bulgaria
georgi@releva.ai

Ivan Koychev
FMI and GATE
Sofia University
Sofia, Bulgaria
koychev@fmi.uni-sofia.bg

Preslav Nakov
Qatar Computing Research Institute
HBKU
Doha, Qatar
pnakov@hbku.edu.qa

Abstract

In education, quiz questions have become an important tool for assessing the knowledge of students. Yet, manually preparing such questions is a tedious task, and thus automatic question generation has been proposed as a possible alternative. So far, the vast majority of research has focused on generating the question text, relying on question answering datasets with readily picked answers, and the problem of how to come up with answer candidates in the first place has been largely ignored. Here, we aim to bridge this gap. In particular, we propose a model that can generate a specified number of answer candidates for a given passage of text, which can then be used by instructors to write questions manually or can be passed as an input to automatic answer-aware question generators. Our experiments show that our proposed answer candidate generation model outperforms several baselines.

1 Introduction

Testing with quiz questions can help both learning and retention, e.g., it could be used for self-study or as a way to detect knowledge gaps in a classroom setting, thus allowing instructors to adapt their teaching (Roediger III et al., 2011).

As creating such quiz questions is a tedious job, automatic methods have been proposed. The task is often formulated as an answer-aware question generation (Heilman and Smith, 2010; Zhang et al., 2014; Du et al., 2017; Du and Cardie, 2018; Sun et al., 2018; Dong et al., 2019; Bao et al., 2020; CH and Saha, 2020): given an input text and a target answer, generate a corresponding question.

Context: *One of the most famous people born in Warsaw was **Maria Skłodowska-Curie**, who achieved international recognition for her research on radioactivity and was the first female recipient of the **Nobel Prize**. **Famous musicians** include Władysław Szpilman and Frédéric Chopin. Though Chopin was born in the village of Żelazowa Wola, about 60 km (37 mi) from Warsaw, he moved to the city with his family when he was **seven months old**. Casimir Pulaski, a Polish general and hero of the American Revolutionary War, was born here in **1745**.*

Q1: *What was Maria Curie the first female recipient of?*
A1: *Nobel Prize*

Q2: *What year was Casimir Pulaski born in Warsaw?*
A2: *1745*

Q3: *How old was Chopin when he moved to Warsaw with his family?*
A3: *seven months old*

Figure 1: Example passage from the SQuAD1.1 dataset showing three of the questions created by humans and all answers to those and other questions in bold.

Many researchers have used the Stanford Question Answering Dataset (SQuAD1.1) dataset (Rajpurkar et al., 2016) as a source of training and testing data for answer-aware question generation. It contains human-generated questions and answers about articles in Wikipedia, as shown in Figure 1.

Proceedings of the Student Research Workshop associated with RANLP-2021, pages 203–209,
held online, Sep 1–3, 2021.
https://doi.org/10.26615/issn.2603-2821.2021_028

However, this formulation requires that answers be picked beforehand, which may not be practical for real-world situations. Here we aim to address this limitation by proposing a method for generating answers, which can in turn serve as an input to answer-aware question generation models. Our model combines orthographic, lexical, syntactic, and semantic information, and shows promising results. It further allows the user to specify the number of answer to propose. Our contributions can be summarized as follows:

- We propose a new task: generate answer candidates that can serve as an input to answer-aware question generation models.
- We create a dataset for this new task.
- We propose a suitable model for the task, which combines orthographic, lexical, syntactic, and semantic information, and can generate a pre-specified number of answers.
- We demonstrate improvements over simple approaches based on named entities, and competitiveness over complex neural models.

2 Related Work

The success of large-scale pre-trained Transformers such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and generative ones such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020), has led to the rise in popularity of the Question Generation task. Models such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2020) and PEGASUS (Zhang et al., 2020) have been used to generate questions for the SQuAD1.1 dataset and have been commonly evaluated (Du et al., 2017) using BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Lavie and Agarwal, 2007). Strong models for this task include NQG++ (Zhou et al., 2017), ProphetNet (Qi et al., 2020), MPQG (Song et al., 2018), UniLM (Dong et al., 2019), UniLMv2 (Bao et al., 2020), and ERNIE-GEN (Xiao et al., 2020).

All these models were trained for answer-aware question generation, which takes the answer and the textual context as an input and outputs a question for that answer. In contrast, our task formulation takes a textual context as an input and generates possible answers; in turn, these answers can be used as an input to the above answer-aware question generation models.

The Quiz-Style Question Generation for News Stories task (Lelkes et al., 2021) uses a formulation that asks to generate a single question as well as the corresponding answer, which is to be extracted from the given context.

Follow-up research has tried to avoid the limitation of generating a single question or a single question–answer pair by generating a question for each sentence in the input context or by using all named entities in the context as answer keys (Montgomerie, 2020).

Finally, there has been a proliferation of educational datasets in recent years (Zeng et al., 2020; Dzendzik et al., 2021; Rogers et al., 2021), which includes Crowdsourcing (Welbl et al., 2017), ARC (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), multiple-choice exams in Bulgarian (Hardalov et al., 2019), Vietnamese (Nguyen et al., 2020), and EXAMS, which covers 16 different languages (Hardalov et al., 2020). Yet, these datasets are not directly applicable for our task as their questions do not expect the answers to be exact matches from the textual context. While there are also span-based extraction datasets such as NewsQA (Trischler et al., 2017), SearchQA (Dunn et al., 2017) and Natural Questions: A Benchmark for Question Answering Research (Kwiatkowski et al., 2019) they contains a mix of long and short spans rather than factoid answers. Thus, we opted to use SQuAD1.1 in our experiments, but focusing on generating answers rather than on questions.

3 Method

Given an input textual context, we first extract phrases from it, then we calculate a representation for each phrase, and finally, we predict which phrases are appropriate of being an answer to a quiz question based on these representations.

3.1 Data

To train our classifier, we need a labeled dataset of key phrases. In particular, we use SQuAD1.1, which consists of more than 100,000 questions created by humans from Wikipedia articles, and was extensively used for question answering. An example is shown in Figure 1. We use version 1.1 of the dataset instead of 2.0 (Rajpurkar et al., 2018) because it contains the exact position of the answers in the text, which allows us to easily match them against the candidate phrases. Version 2.0 only adds examples whose answer is not present.

We created a dataset for our task using 87,600 questions from the SQuAD1.1 training set and their associated textual contexts. Because only 33% of the answers consisted of one word, it is important to also extract phrases longer than a single word. Thus, we also added all named entities; note that they have a variable word length. We further included all noun chunks, which we then extended by combining two or more noun chunks if the only words between them were connectors like *and*, *of*, and *or*. Here is an example of a complex chunk with three pieces: *a Marian place* ***of*** *prayer* ***and*** *reflection*. We considered as positive examples the phrases for which there was a question asked in the SQuAD1.1 dataset, and we considered as negative examples the additional phrases we created.

3.2 Features

We extracted the following features, adapted for the use of phrases containing multiple words:

TFIDFArticle, **TFIDFParagraph**: The average TF.IDF score for all words in the key phrase, where the Inverse Document Frequency (IDF) is computed from the words in all paragraphs of the article (*TFIDFArticle*) or only from the paragraph of the given key phrase (*TFIDFParagraph*).

TitleSimilarity: The average cosine similarity between the vectors of the words in the key phrase and the article title.

POS, **TAG**, **DEP**: The coarse-grained part-of-speech tag (*POS*), the fine-grained part-of-speech tag (*TAG*), and the syntactic dependency relation (*DEP*). If the phrase contains multiple words, we only consider the word with the highest TF.IDF.

EntityType: The named entity type of the phrase if any.

IsAlpha: True if all characters in the phrase are alphabetic.

IsAscii: True if the phrase consists only of characters contained in the standard ASCII table.

IsDigit: True if the phrase only contains digits.

IsLower: True if all words in the phrase are in lowercase.

IsCapital: True if the first word in the phrase is in uppercase.

IsCurrency: True if some word in the phrase contains a currency symbol, e.g., *$23*.

LikeNum: True if some word in the phrase represents a number, e.g., *13.4*, *42*, *twenty*, etc.

3.3 Model

We convert all the above features to binary, and then we use a Bernoulli Naïve Bayes classifier, which can account both for the presence and for the absence of a feature. To achieve this, we encode categorical features (e.g., *POS*, *TAG*) using one-hot encoding, and we put continuous features (e.g., *TFIDFArticle*, *TitleSimilarity*) into bins of five.

3.4 Evaluation Measures

As there is no established measure for evaluating key phrases for answer generation, we use and adapt the original evaluation script[1] created for the Question Answering task on the SQuAD1.1 dataset (Rajpurkar et al., 2016), which calculates an exact match (*EM*) and the harmonic mean of precision and recall (*F1*).

In the SQuAD1.1 dataset, there can be multiple correct versions of the answer for a question (e.g., *third*, *third-most*). Thus, the evaluation script calculates EM and F1 for each such version and then returns the highest value. As there can be also multiple question–answer pairs in a given passage, we further adapted the script to include all human-created answers, we calculated these scores against all answers in the passage, and finally, we took the highest values.

Finally, in order to allow for a more practical use of question generation algorithms, it is desirable to be able to generate multiple question–answer pairs for a given passage. To compute EM and F1 over multiple answer candidates, we adopted the following two approaches:

EM-Any and **F1-Any** show how likely it is to pick a ground-truth answer (also, how likely it is to be chosen by a human annotator of SQuAD1.1) out of *N* returned candidate answers. To calculate them, we took only the best EM and F1 scores after computing all scores for each candidate answer.

Using **EM-Avg** and **F1-Avg**, we can measure what percentage of all returned candidate answers have also been marked as an answer by a human. To calculate them, we took the average of all EM and F1 scores computed for the proposed candidate answers.

The results for the SQuAD1.1 development split, which consists of 2,067 unique passages, are shown in Table 1 and Table 2.

[1] http://github.com/allenai/bi-att-flow/blob/master/squad/evaluate-v1.1.py

4 Experiments and Evaluation

We used our model to generate ten candidate answers per passage (taking the ones with the highest classifier confidence), and we compared the results to other commonly used methods.

4.1 Baselines

Below, we list the baselines that we compared against:

- **NER:** Extracting all named entities from the passage and using them as candidate answers. On average, there are 13.64 named entities per SQuAD1.1 passage.
- **Noun Chunks:** Extracting all noun chunks from the passage and using them as candidate answers. On average, there are 33.15 noun chunks per SQuAD1.1 passage.
- **NE + NCh:** Combining all extracted named entities and noun chunks from the passage after using the SQuAD1.1 normalization script[2] to remove duplicate words (e.g., *the third* matches *third*).
- **T5-small:** We fine-tuned the small version of T5, which has 220M parameters. We trained the model to accept the passage as an input and to output the answer. We used a learning rate of 0.0001, a source token length of 300, and a target token length of 24. The best validation loss was achieved in the forth out of ten epochs.

4.2 Results

In this section, we describe our experimental results and we compare them to the baselines described in Section 4.1 above.

4.2.1 Best Over Multiple Candidates

Table 1 shows the results for EM-Any and F1-Any, i.e., how often, among the top-N candidates by the model, at least one was picked by a human.

We can see that, compared to using named entities, our model achieves a better EM-Any score with just eight answer candidates rather than using all named entities in the passage (which are 13.6 on average). It also achieves a higher F1-Any score with just six answer candidates.

Method	Answers	EM-Any	F1-Any
Our Model	1	29.63	38.80
	2	42.50	55.47
	3	52.92	66.83
	4	59.67	73.92
	5	65.50	79.11
	6	69.33	82.50
	7	72.90	85.58
	8	**75.43**	**87.70**
	9	77.66	89.20
	10	79.17	90.32
NER	**13.6**	**74.36**	**82.12**
NCh	33.2	86.79	95.90
NER + NCh	35.4	95.02	98.48
T5-small	1	37.56	49.16

Table 1: **Best over multiple candidates (EM-Any and F1-Any).** Measuring how often, among the top-N candidates proposed by the model, at least one was picked by a human.

Method	Answers	EM-Avg	F1-Avg
Our Model	1	**29.63**	**38.80**
	2	25.58	36.03
	3	24.18	35.15
	4	22.74	34.15
	5	22.07	33.65
	6	20.90	32.81
	7	20.02	32.05
	8	19.25	31.43
	9	18.45	30.57
	10	17.64	29.75
NER	13.6	16.33	25.24
NCh	33.2	7.86	17.75
NER + NCh	35.4	8.97	18.84
T5-small	1	**37.56**	**49.16**

Table 2: **Average over multiple candidates (EM-Avg, F1-Avg).** Measuring what percentage of the proposed answers were also selected as an answer by a human.

We further see that using the combination of all named entities and noun chunks yields the best score, but it produces 35 candidates on average, which is the majority of the words in the passage.

4.2.2 Average Over Multiple Candidates

Table 2 shows the results for EM-Avg and F1-Avg, i.e., measuring what percentage of the proposed answers were also selected as an answer by a human.

[2] http://github.com/allenai/bi-att-flow/blob/master/squad/evaluate-v1.1.py#L11

Due to the ability of the classifier to take a lower number of candidate questions, we can see that it outperforms taking all named entities or all noun chunks by a sizable margin.

We further see that the average scores consistently drop with the increase of the number of answer candidates. This also explains the lower scores of the named entity and noun chunks approaches as they produce much longer lists of candidate answers.

4.2.3 Single Answer Candidate

Finally, we see in both tables, that the T5 model achieves the highest average result. However, in our setup it cannot produce multiple candidates. We plan to extend it accordingly in future work.

5 Discussion

Figure 2 shows a passage from the development split of the SQuAD1.1 dataset and the top-10 answers that our model proposed for it. We can see that these answers represent a diverse set, including named entities, noun chunks, and individual words. Indeed, this is a typical example, as our analysis across the entire development dataset shows that on average, among the top-10 candidates, our model proposes 4.82 named entities and 6.40 noun chunks.

Note also that our evaluation setup could be unfair to the model in some cases, e.g., if the model proposes a good candidate answer but one that was not chosen by the human annotators, it would receive no credit for it.

Finally, note that our model can produce top-N results for user-defined values of N, which is an advantage over simple baselines based on entities or chunks, as well as over our setup for T5.

6 Conclusion and Future Work

We proposed a new task: generate answer candidates that can serve as an input to answer-aware question generation models. We further created a dataset for this new task. Moreover, we proposed a suitable model for the task, which combines orthographic, lexical, syntactic, and semantic information, and can generate a pre-specified number of answers. Finally, we demonstrated improvements over simple approaches based on named entities, and competitiveness over complex, computationally expensive neural network models such as T5.

Context: *Oxygen is a chemical element with symbol O and atomic number 8. It is a member of the chalcogen group on the periodic table and is a highly reactive nonmetal and oxidizing agent that readily forms compounds (notably oxides) with most elements. By mass, oxygen is the third-most abundant element in the universe, after hydrogen and helium. At standard temperature and pressure, two atoms of the element bind to form dioxygen, a colorless and odorless diatomic gas with the formula O2. Diatomic oxygen gas constitutes 20.8% of the Earth's atmosphere. However, monitoring of atmospheric oxygen levels show a global downward trend, because of fossil-fuel burning. Oxygen is the most abundant element by mass in the Earth's crust as part of oxide compounds such as silicon dioxide, making up almost half of the crust's mass.*

Top 10 answers:

1. 8
2. 20.8 % of the Earth 's atmosphere
3. Oxygen
4. a member of the chalcogen group
5. Diatomic oxygen gas
6. almost half
7. O
8. two atoms of the element
9. the formula O2
10. fossil-fuel burning

Figure 2: The top-10 answer candidates generated by our model for a sample passage from SQuAD1.1. The human-selected ground truth answers are underlined, and the answer candidates are shown in brown.

In future work, we plan to analyze and to improve the features. We also want to extend T5 to generate multiple candidates. We further plan to reduce the impact of false negatives, e.g., by means of manual evaluation by domain experts, and eventually by producing datasets with (potentially ranked) annotations of all suitable candidate answers.

Acknowledgments

This research is partially funded via Project UNITe by the OP "Science and Education for Smart Growth" and co-funded by the EU through the ESI Funds under GA No. BG05M2OP001-1.001-0004.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652.

Dhawaleswar Rao CH and Sujan Kumar Saha. 2020. Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies*, 13(1):14–25.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv:1803.05457*.

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*, NAACL-HLT '19, pages 4171–4186, Minneapolis, Minnesota, USA.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, NeurIPS '19, pages 13042–13054, Vancouver, BC, Canada.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL '18, pages 1907–1917, Melbourne, Australia.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 1342–1352, Vancouver, Canada.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv:1704.05179*.

Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2021. English machine reading comprehension datasets: A survey. *arXiv:2101.10421*.

Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019. Beyond English-only reading comprehension: Experiments in zero-shot multilingual transfer for Bulgarian. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '19, pages 447–459, Varna, Bulgaria.

Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov. 2020. EXAMS: A multi-subject high school examinations dataset for cross-lingual and multilingual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 5427–5444, Online.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '10, pages 609–617, Los Angeles, California, USA. 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.

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. In *Proceedigs of the 8th International Conference on Learning Representations*, ICLR '20, Addis Ababa, Ethiopia.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, WMT '07, pages 228–231, Prague, Czech Republic.

Adam D. Lelkes, Vinh Q. Tran, and Cong Yu. 2021. Quiz-style question generation for news stories. In *Proceedings of the Web Conference 2021*, WWW '21, pages 2501–2511.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 7871–7880.

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

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:1907.11692*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 2381–2391, Brussels, Belgium.

Adam Montgomerie. 2020. Question generator. https://github.com/AMontgomerie/question_generator.

Kiet Van Nguyen, Khiem Vinh Tran, Son T. Luu, Anh Gia-Tuan Nguyen, and Ngan Luu-Thuy Nguyen. 2020. Enhancing lexical-based approach with external knowledge for Vietnamese multiple-choice machine reading comprehension. *IEEE Access*, 8:201404–201417.

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*, ACL '02, pages 311–318, Philadelphia, Pennsylvania, USA.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL '18, pages 784–789, Melbourne, Australia.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, pages 2383–2392, Austin, Texas.

Henry L. Roediger III, Adam L. Putnam, and Megan A. Smith. 2011. Chapter one - ten benefits of testing and their applications to educational practice. In Jose P. Mestre and Brian H. Ross, editors, *Psychology of Learning and Motivation*, volume 55, pages 1–36. Academic Press.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension. *arXiv:2107.12708*.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '18, pages 569–574, New Orleans, Louisiana, USA.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 3930–3939, Brussels, Belgium.

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*, RepL4NLP '17, pages 191–200, Vancouver, Canada.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, W-NUT '17, pages 94–106, Copenhagen, Denmark.

Dongling Xiao, Han Zhang, Yu-Kun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-GEN: an enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI '20, pages 3997–4003.

Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Applied Sciences*, 10(21).

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, pages 11328–11339.

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, page 371–380, Shanghai, China.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. *arXiv:1704.01792*.