

# Simple and Effective Semi-Supervised Question Answering

Bhuwan Dhingra\*    Danish Pruthi\*    Dheeraj Rajagopal\*

School of Computer Science

Carnegie Mellon University, Pittsburgh, USA

{bdhingra, ddanish, dheeraj}@cs.cmu.edu

## Abstract

Recent success of deep learning models for the task of extractive Question Answering (QA) is hinged on the availability of large annotated corpora. However, large domain specific annotated corpora are limited and expensive to construct. In this work, we envision a system where the end user specifies a set of base documents and only a few labelled examples. Our system exploits the document structure to create cloze-style questions from these base documents; pre-trains a powerful neural network on the cloze style questions; and further fine-tunes the model on the labeled examples. We evaluate our proposed system across three diverse datasets from different domains, and find it to be highly effective with very little labeled data. We attain more than 50% F1 score on SQuAD and TriviaQA with less than a thousand labelled examples. We are also releasing a set of 3.2M cloze-style questions for practitioners to use while building QA systems<sup>1</sup>.

## 1 Introduction

Deep learning systems have shown a lot of promise for extractive Question Answering (QA), with performance comparable to humans when large scale data is available. However, practitioners looking to build QA systems for specific applications may not have the resources to collect tens of thousands of questions on corpora of their choice. At the same time, state-of-the-art machine reading systems do not lend well to low-resource QA settings where the number of labeled question-answer pairs are limited (c.f. Table 2). Semi-supervised QA methods like (Yang et al., 2017) aim to improve this performance by leveraging unlabeled data which is easier to collect.

In this work, we present a semi-supervised QA system which requires the end user to specify a

set of base documents and only a small set of question-answer pairs over a subset of these documents. Our proposed system consists of three stages. First, we construct cloze-style questions (predicting missing spans of text) from the unlabeled corpus; next, we use the generated clozes to pre-train a powerful neural network model for extractive QA (Clark and Gardner, 2017; Dhingra et al., 2017); and finally, we fine-tune the model on the small set of provided QA pairs.

Our cloze construction process builds on a typical writing phenomenon and document structure: an introduction precedes and summarizes the main body of the article. Many large corpora follow such a structure, including Wikipedia, academic papers, and news articles. We hypothesize that we can benefit from the un-annotated corpora to better answer various questions – at least ones that are lexically similar to the content in base documents and directly require factual information.

We apply the proposed system on three datasets from different domains – SQuAD (Rajpurkar et al., 2016), TriviaQA-Web (Joshi et al., 2017) and the BioASQ challenge (Tsatsaronis et al., 2015). We observe significant improvements in a low-resource setting across all three datasets. For SQuAD and TriviaQA, we attain an F1 score of more than 50% by merely using 1% of the training data. Our system outperforms the approaches for semi-supervised QA presented in Yang et al. (2017), and a baseline which uses the same unlabeled data but with a language modeling objective for pretraining. In the BioASQ challenge, we outperform the best performing system from previous year’s challenge, improving over a baseline which does transfer learning from the SQuAD dataset. Our analysis reveals that questions which ask for factual information and match to specific parts of the context documents benefit the most from pre-training on automatically constructed clozes.

\*Equal Contribution

<sup>1</sup><http://bit.ly/semi-supervised-qa>

## 2 Related Work

**Semi-supervised learning** augments the labeled dataset  $L$  with a potentially larger unlabeled dataset  $U$ . Yang et al. (2017) presented a model, GDAN, which trained an auxiliary neural network to generate questions from passages by reinforcement learning, and augment the labeled dataset with the generated questions to train the QA model. Here we use a much simpler heuristic to generate the auxiliary questions, which also turns out to be more effective as we show superior performance compared to GDAN. Several approaches have been suggested for generating natural questions (Tang et al., 2017; Subramanian et al., 2017; Song et al., 2017), however none of them show a significant improvement of using the generated questions in a semi-supervised setting. Recent papers also use unlabeled data for QA by training large language models and extracting contextual word vectors from them to input to the QA model (Salant and Berant, 2017; Peters et al., 2018; McCann et al., 2017). The applicability of this method in the low-resource setting is unclear as the extra inputs increase the number of parameters in the QA model, however, our pretraining can be easily applied to these models as well.

**Domain adaptation** (and **Transfer learning**) leverage existing large scale datasets from a source domain (or task) to improve performance on a target domain (or task). For deep learning and QA, a common approach is to pretrain on the source dataset and then fine-tune on the target dataset (Chung et al., 2017; Golub et al., 2017). Wiese et al. (2017) used SQuAD as a source for the target BioASQ dataset, and Kadlec et al. (2016) used Book Test (Bajjar et al., 2016) as source for the target SQuAD dataset. Mihaylov et al. (2017) transfer learned model layers from the tasks of sequence labeling, text classification and relation classification to show small improvements on SQuAD. All these works use manually curated source dataset, which in themselves are expensive to collect. Instead, we show that it is possible to automatically construct the source dataset from the same domain as the target, which turns out to be more beneficial in terms of performance as well (c.f. Section 4). Several cloze datasets have been proposed in the literature which use heuristics for construction (Hermann et al., 2015; Onishi et al., 2016; Hill et al., 2016). We further see the usability of such a dataset in a semi-supervised setting.

## 3 Methodology

Our system comprises of following three steps:

**Cloze generation:** Most of the documents typically follow a template, they begin with an introduction that provides an overview and a brief summary for what is to follow. We assume such a structure while constructing our cloze style questions. When there is no clear demarcation, we treat the first  $K\%$  (hyperparameter, in our case 20%) of the document as the introduction. While noisy, this heuristic generates a large number of clozes given any corpus, which we found to be beneficial for semi-supervised learning despite the noise.

We use a standard NLP pipeline based on Stanford CoreNLP<sup>2</sup> (for SQuAD, TrivaQA and PubMed) and the BANNER Named Entity Recognizer<sup>3</sup> (only for PubMed articles) to identify entities and phrases. Assume that a document comprises of introduction sentences  $\{q_1, q_2, \dots, q_n\}$ , and the remaining passages  $\{p_1, p_2, \dots, p_m\}$ . Additionally, let's say that each sentence  $q_i$  in introduction is composed of words  $\{w_1, w_2, \dots, w_{l_{q_i}}\}$ , where  $l_{q_i}$  is the length of  $q_i$ . We consider a match( $q_i, p_j$ ), if there is an exact string match of a sequence of words  $\{w_k, w_{k+1}, \dots, w_{l_{q_i}}\}$  between the sentence  $q_i$  and passage  $p_j$ . If this sequence is either a noun phrase, verb phrase, adjective phrase or a named entity in  $p_j$ , as recognized by CoreNLP or BANNER, we select it as an answer span  $A$ . Additionally, we use  $p_j$  as the passage  $P$  and form a cloze question  $Q$  from the answer bearing sentence  $q_i$  by replacing  $A$  with a placeholder. As a result, we obtain passage-question-answer ( $P, Q, A$ ) triples (Table 1 shows an example). As a post-processing step, we prune out ( $P, Q, A$ ) triples where the word overlap between the question ( $Q$ ) and passage ( $P$ ) is less than 2 words (after excluding the stop words).

<i>Passage (P)</i> : Autism is a neurodevelopmental disorder characterized by impaired <b>social interaction</b> , verbal and non-verbal communication, and ...
<i>Question (Q)</i> : People with autism tend to be a little aloof with little to no _____.
<i>Answer (A)</i> : social interaction

Table 1: An example constructed cloze.

The process relies on the fact that answer candidates from the introduction are likely to be discussed in detail in the remainder of the article.

<sup>2</sup><https://stanfordnlp.github.io/CoreNLP/>

<sup>3</sup><http://banner.sourceforge.net>

In effect, the cloze question from the introduction and the matching paragraph in the body forms a question and context passage pair. We create two cloze datasets, one each from Wikipedia corpus (for SQuAD and TriviaQA) and PubMed academic papers (for the BioASQ challenge), consisting of 2.2M and 1M clozes respectively. From analyzing the cloze data manually, we were able to answer 76% times for the Wikipedia set and 80% times for the PubMed set using the information in the passage. In most cases the cloze paraphrased the information in the passage, which we hypothesized to be a useful signal for the downstream QA task.

We also investigate the utility of forming subsets of the large cloze corpus, where we select the top passage-question-answer triples, based on the different criteria, like i) jaccard similarity of answer bearing sentence in introduction and the passage ii) the tf-idf scores of answer candidates and iii) the length of answer candidates. However, we empirically find that we were better off using the entire set rather than these subsets.

**Pre-training:** We make use of the generated cloze dataset to pre-train an expressive neural network designed for the task of reading comprehension. We work with two publicly available neural network models – the GA Reader (Dhingra et al., 2017) (to enable comparison with prior work) and BiDAF + Self-Attention (SA) model from Clark and Gardner (2017) (which is among the best performing models on SQuAD and TriviaQA). After pretraining, the performance of BiDAF+SA on a dev set of the (Wikipedia) cloze questions is 0.58 F1 score and 0.55 Exact Match (EM) score. This implies that the cloze corpus is neither too easy, nor too difficult to answer.

**Fine Tuning:** We fine tune the pre-trained model, from the previous step, over a small set of labelled question-answer pairs. As we shall later see, this step is crucial, and it only requires a handful of labelled questions to achieve a significant proportion of the performance typically attained by training on tens of thousands of questions.

## 4 Experiments & Results

### 4.1 Datasets

We apply our system to three datasets from different domains. **SQuAD** (Rajpurkar et al., 2016) consists of questions whose answers are free form spans of text from passages in Wikipedia articles.

We follow the same setting as in (Yang et al., 2017), and split 10% of training questions as the test set, and report performance when training on subsets of the remaining data ranging from 1% to 90% of the full set. We also report the performance on the dev set when trained on the full training set (1\* in Table 2). We use the same hyperparameter settings as in prior work. We compare and study four different settings: 1) the Supervised Learning (SL) setting, which is only trained on the supervised data, 2) the best performing **GDAN** model from Yang et al. (2017), 3) pretraining on a Language Modeling (LM) objective and fine-tuning on the supervised data, and 4) pretraining on the **Cloze** dataset and fine-tuning on the supervised data. The LM and Cloze methods use exactly the same data for pretraining, but differ in the loss functions used. We report F1 and EM scores on our test set using the official evaluation scripts provided by the authors of the dataset.

**TriviaQA** (Joshi et al., 2017) comprises of over 95K web question-answer-evidence triples. Like SQuAD, the answers are spans of text. Similar to the setting in SQuAD, we create multiple smaller subsets of the entire set. For our semi-supervised QA system, we use the BiDAF+SA model (Clark and Gardner, 2017) – the highest performing publicly available system for TriviaQA. Here again, we compare the supervised learning (SL) settings against the pretraining on **Cloze** set and fine tuning on the supervised set. We report F1 and EM scores on the dev set<sup>4</sup>.

We also test on the **BioASQ 5b dataset**, which consists of question-answer pairs from PubMed abstracts. We use the publicly available system<sup>5</sup> from Wiese et al. (2017), and follow the exact same setup as theirs, focusing only on factoid and list questions. For this setting, there are only 899 questions for training. Since this is already a low-resource problem we only report results using 5-fold cross-validation on all the available data. We report Mean Reciprocal Rank (MRR) on the factoid questions, and F1 score for the list questions.

### 4.2 Main Results

Table 2 shows a comparison of the discussed settings on both SQuAD and TriviaQA. Without any

<sup>4</sup>We use a sample of dev questions, which is the default setting for the code by Clark and Gardner (2017). Since our goal is only to compare the models, this is not problematic.

<sup>5</sup><https://github.com/georgwiese/biomedical-qa>

Model	Method	0		0.01		0.05		0.1		0.2		0.5		0.9		1	
		F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
SQuAD																	
GA	SL	-	-	0.0882	0.0359	0.3517	0.2275	0.4116	0.2752	0.4797	0.3393	0.5705	0.4224	0.6125	0.4684	-	-
GA	GDAN	-	-	-	-	-	-	0.4840	0.3270	0.5394	0.3781	0.5831	0.4267	0.6102	0.4531	-	-
GA	LM	-	-	0.0957	0.0394	0.3141	0.1856	0.3725	0.2365	0.4406	0.2983	0.5111	0.3589	0.5520	0.3964	-	-
GA	Cloze	-	-	0.3090	0.1964	0.4688	0.3385	0.4937	0.3588	0.5575	0.4126	0.6086	0.4679	0.6302	0.4894	-	-
BiDAF+SA	SL	-	-	0.1926	0.1018	0.4764	0.3388	0.5639	0.4258	0.6484	0.5031	0.7044	0.5615	0.7287	0.5874	0.8069	0.7154
BiDAF+SA	Cloze	<b>0.0682</b>	<b>0.032</b>	<b>0.5042</b>	<b>0.3751</b>	<b>0.6324</b>	<b>0.4862</b>	<b>0.6431</b>	<b>0.4995</b>	<b>0.6839</b>	<b>0.5413</b>	<b>0.7151</b>	<b>0.5767</b>	<b>0.7369</b>	<b>0.6005</b>	<b>0.8080</b>	<b>0.7186</b>
TRIVIA-QA																	
BiDAF+SA	SL	-	-	0.2533	0.1898	0.4215	0.3566	0.4971	0.4318	0.5624	0.5077	0.6867	0.6239	0.7131	0.6617	0.7291	0.6786
BiDAF+SA	Cloze	<b>0.1182</b>	<b>0.0729</b>	<b>0.5521</b>	<b>0.4807</b>	<b>0.6245</b>	<b>0.5614</b>	<b>0.6506</b>	<b>0.5893</b>	<b>0.6849</b>	<b>0.6281</b>	<b>0.7196</b>	<b>0.6607</b>	<b>0.7381</b>	<b>0.6823</b>	<b>0.7461</b>	<b>0.6903</b>

Table 2: A holistic view of the performance of our system compared against baseline systems on SQuAD and TriviaQA. Column groups represent different fractions of the training set used for training.

fine-tuning (column 0) the performance is low, probably because the model never saw a real question, but we see significant gains with Cloze pretraining even with very little labeled data. The BiDAF+SA model, exceeds an F1 score of 50% with only 1% of the training data (454 questions for SQuAD, and 746 questions for TriviaQA), and approaches 90% of the best performance with only 10% labeled data. The gains over the SL setting, however, diminish as the size of the labeled set increases and are small when the full dataset is available.

Method	Factoid MRR	List F1
SL*	0.242	0.211
SQuAD pretraining	0.262	0.211
Cloze pretraining	<b>0.328</b>	<b>0.230</b>

Table 3: 5-fold cross-validation results on BioASQ Task 5b. \*Our SL experiments showed better performance than what was reported in (Wiese et al., 2017).

Cloze pretraining outperforms the GDAN baseline from Yang et al. (2017) using the same SQuAD dataset splits. Additionally, we show improvements in the 90% data case unlike GDAN. Our approach is also applicable in the extremely low-resource setting of 1% data, which we suspect GDAN might have trouble with since it uses the labeled data to do reinforcement learning. Furthermore, we are able to use the same cloze dataset to improve performance on both SQuAD and TriviaQA datasets. When we use the same unlabeled data to pre-train with a language modeling objective, the performance is worse<sup>6</sup>, showing the bias we introduce by constructing clozes is important.

<sup>6</sup>Since the GA Reader uses *bidirectional* RNN layers, when pretraining the LM we had to mask the inputs to the intermediate layers partially to avoid the model being exposed to the labels it is predicting. This results in only a subset of the parameters being pretrained, which is why we believe this baseline performs poorly.

On the BioASQ dataset (Table 3) we again see a significant improvement when pretraining with the cloze questions over the supervised baseline. The improvement is smaller than what we observe with SQuAD and TriviaQA datasets – we believe this is because questions are generally more difficult in BioASQ. Wiese et al. (2017) showed that pretraining on SQuAD dataset improves the downstream performance on BioASQ. Here, we show a much larger improvement by pretraining on cloze questions constructed in an *unsupervised* manner from the same domain.

### 4.3 Analysis

**Regression Analysis:** To understand which types of questions benefit from pre-training, we pre-specified certain features (see Figure 1 right) for each of the dev set questions in SQuAD, and then performed linear regression to predict the F1 score for that question from these features. We predict the F1 scores from the cloze pretrained model ( $y^{\text{cloze}}$ ), the supervised model ( $y^{\text{sl}}$ ), and the difference of the two ( $y^{\text{cloze}} - y^{\text{sl}}$ ), when using 10% of labeled data. The coefficients of the fitted model are shown in Figure 1 (left) along with their std errors. Positive coefficients indicate that a high value of that feature is predictive of a high F1 score, and a negative coefficient indicates that a small value of that feature is predictive of a high F1 score (or a high difference of F1 scores from the two models in the case of  $y^{\text{cloze}} - y^{\text{sl}}$ ).

The two strongest effects we observe are that a high lexical overlap between the question and the sentence containing the answer is indicative of high boost with pretraining, and that a high lexical overlap between the question and the whole passage is indicative of the opposite. This is hardly surprising, since our cloze construction process is biased towards questions which have a



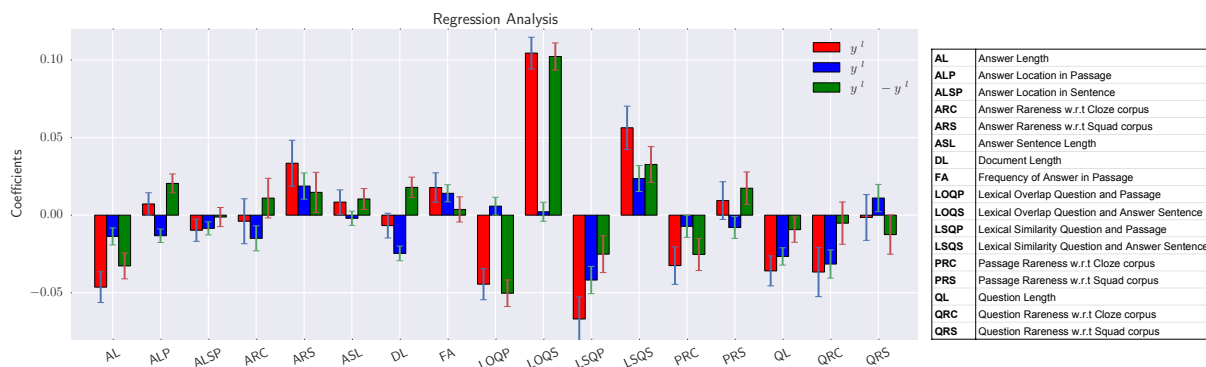


Figure 1: **Left:** Regression coefficients, along with std-errors, when predicting F1 score of *cloze* model, or *sl* model, or the difference of the two, from features computed from SQuAD dev set questions. **Right:** Descriptions of the features.

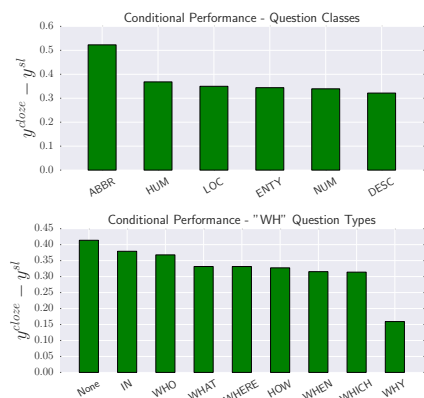


Figure 2: Performance gain with pretraining for different subsets of question types.

similar phrasing to the answer sentences in context. Hence, test questions with a similar property are answered correctly after pretraining, whereas those with a high overlap with the whole passage tend to have lower performance. The pretraining also favors questions with short answers because the cloze construction process produces short answer spans. Also passages and questions which consist of tokens infrequent in the SQuAD training corpus receive a large boost after pretraining, since the unlabeled data covers a larger domain.

**Performance on question types:** Figure 2 shows the average gain in F1 score for different types of questions, when we pretrain on the clozes compared to the supervised case. This analysis is done on the 10% split of the SQuAD training set. We consider two classifications of each question – one determined on the first word (usually a wh-word) of the question (Figure 2 (bottom)) and one based on the output of a separate question type classifier<sup>7</sup> adapted from (Li and Roth,

<sup>7</sup><https://github.com/brmson/question-classification>

2002). We use the coarse grain labels namely Abbreviation (ABBR), Entity (ENTY), Description (DESC), Human (HUM), Location (LOC), Numeric (NUM) trained on a Logistic Regression classification system. While there is an improvement across the board, we find that abbreviation questions in particular receive a large boost. Also, "why" questions show the least improvement, which is in line with our expectation, since these usually require reasoning or world knowledge which cloze questions rarely require.

## 5 Conclusion

In this paper, we show that pre-training QA models with automatically constructed cloze questions improves the performance of the models significantly, especially when there are few labeled examples. The performance of the model trained only on the cloze questions is poor, validating the need for fine-tuning. Through regression analysis, we find that pretraining helps with questions which ask for factual information located in a specific part of the context. For future work, we plan to explore the active learning setup for this task – specifically, which passages and / or types of questions can we select to annotate, such that there is a maximum performance gain from fine-tuning. We also want to explore how to adapt cloze style pretraining to NLP tasks other than QA.

## Acknowledgments

Bhuwan Dhingra is supported by NSF under grants CCF-1414030 and IIS-1250956 and by grants from Google. Danish Pruthi and Dheeraj Rajagopal are supported by the DARPA Big Mechanism program under ARO contract W911NF-14-1-0436.

## References

- Ondrej Bajgar, Rudolf Kadlec, and Jan Kleindienst. 2016. Embracing data abundance: Booktest dataset for reading comprehension. *arXiv preprint arXiv:1610.00956*.
- Yu-An Chung, Hung-Yi Lee, and James Glass. 2017. Supervised and unsupervised transfer learning for question answering. *arXiv preprint arXiv:1711.05345*.
- Christopher Clark and Matt Gardner. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.
- Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. *ACL*.
- David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. *arXiv preprint arXiv:1706.09789*.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.
- Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children’s books with explicit memory representations. *ICLR*.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.
- Rudolf Kadlec, Ondřej Bajgar, Peter Hrinčar, and Jan Kleindienst. 2016. Finding a jack-of-all-trades: An examination of semi-supervised learning in reading comprehension.
- Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *arXiv preprint arXiv:1708.00107*.
- Todor Mihaylov, Zornitsa Kozareva, and Anette Frank. 2017. Neural skill transfer from supervised language tasks to reading comprehension. *arXiv preprint arXiv:1711.03754*.
- Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. *EMNLP*.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *EMNLP*.
- Shimi Salant and Jonathan Berant. 2017. Contextualized word representations for reading comprehension. *arXiv preprint arXiv:1712.03609*.
- Lin Feng Song, Zhiguo Wang, and Wael Hamza. 2017. A unified query-based generative model for question generation and question answering. *arXiv preprint arXiv:1709.01058*.
- Sandeep Subramanian, Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. Neural models for key phrase detection and question generation. *arXiv preprint arXiv:1706.04560*.
- Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.
- George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.
- Georg Wiese, Dirk Weissenborn, and Mariana L. Neves. 2017. [Neural question answering at bioasq 5b](#). In *BioNLP 2017, Vancouver, Canada, August 4, 2017*, pages 76–79.
- Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. *ACL*.