Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering

Alexander R. Fabbri† [1,2] **Patrick Ng**‡[1]
Zhiguo Wang‡ **Ramesh Nallapati**‡ **Bing Xiang**‡
[†]Yale University [‡]AWS AI Labs
alexander.fabbri@yale.edu, {patricng, zhiguow, rnallapa, bxiang}@amazon.com

Abstract

Question Answering (QA) is in increasing demand as the amount of information available online and the desire for quick access to this content grows. A common approach to QA has been to fine-tune a pretrained language model on a task-specific labeled dataset. This paradigm, however, relies on scarce, and costly to obtain, large-scale human-labeled data. We propose an unsupervised approach to training QA models with generated pseudo-training data. We show that generating questions for QA training by applying a simple template on a related, retrieved sentence rather than the original context sentence improves downstream QA performance by allowing the model to learn more complex context-question relationships. Training a QA model on this data gives a relative improvement over a previous unsupervised model in F1 score on the SQuAD dataset by about 14%, and 20% when the answer is a named entity, achieving state-of-the-art performance on SQuAD for unsupervised QA.

1 Introduction

Question Answering aims to answer a question based on a given knowledge source. Recent advances have driven the performance of QA systems to above or near-human performance on QA datasets such as SQuAD (Rajpurkar et al., 2016) and Natural Questions (Kwiatkowski et al., 2019) thanks to pretrained language models such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019). Fine-tuning these language models, however, requires large-scale data for fine-tuning. Creating a dataset for every new domain is extremely costly and practically infeasible. The ability to apply QA models on out-of-domain data in an efficient manner is thus very desirable. This problem may be approached with domain adaptation or transfer learning techniques (Chung et al., 2018) as well as data augmentation (Yang et al., 2017; Dhingra et al., 2018; Wang et al., 2018; Alberti et al., 2019). However, here we expand upon the recently introduced task of unsupervised question answering (Lewis et al., 2019) to examine the extent to which synthetic training data alone can be used to train a QA model.

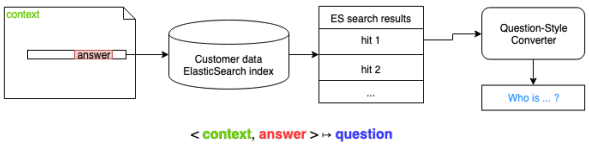

Figure 1: Question Generation Pipeline: the original context sentence containing a given answer is used as a query to retrieve a related sentence containing matching entities, which is input into our question-style converter to create QA training data.

In particular, we focus on the machine reading comprehension setting in which the context is a given paragraph, and the QA model can only access this paragraph to answer a question. Furthermore, we work on extractive QA, where the answer is assumed to be a contiguous sub-string of the context. A training instance for supervised reading comprehension consists of three components: a *question*, a *context*, and an *answer*. For a given dataset domain, a collection of documents can usually be easily obtained, providing *context* in the form of paragraphs or sets of sentences. *Answers* can be gathered from keywords and phrases from the context. We focus mainly on factoid QA; the question concerns a concise fact. In particular, we emphasize questions whose answers are named entities, the majority type of factoid questions. Entities can be extracted from text using named entity recognition (NER) techniques as the training instance's *answer*. Thus, the main challenge, and the focus

[1]Equal contribution
[2]Work done during internship at the AWS AI Labs

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4508–4513
July 5 - 10, 2020.

of this paper, is creating a relevant *question* from a *(context, answer)* pair in an unsupervised manner.

Recent work of (Lewis et al., 2019) uses style transfer for generating questions for *(context, answer)* pairs but shows little improvement over applying a much simpler question generator which drops, permutates and masks words. We improve upon this paper by proposing a simple, intuitive, **retrieval and template-based question generation** approach, illustrated in Figure 1. The idea is to retrieve a sentence from the corpus similar to the current context, and then generate a *question* based on that sentence. Having created a *question* for all *(context, answer)* pairs, we then fine-tune a pre-trained BERT model on this data and evaluate on the SQuAD v1.1 dataset (Rajpurkar et al., 2016).

Our contributions are as follows: we introduce a retrieval, template-based framework which achieves state-of-the-art results on SQuAD for unsupervised models, particularly when the answer is a named entity. We perform ablation studies to determine the effect of components in template question generation. We are releasing our synthetic training data and code.[1]

2 Unsupervised QA Approach

We focus on creating high-quality, non-trivial questions which will allow the model to learn to extract the proper answer from a context-question pair.

Sentence Retrieval: A standard cloze question can be obtained by taking the original sentence in which the answer appears from the context and masking the answer with a chosen token. However, a model trained on this data will only learn text matching and how to fill-in-the-blank, with little generalizability. For this reason, we chose to use a retrieval-based approach to obtain a sentence similar to that which contains the answer, upon which to create a given question. For our experiments, we focused on answers which are named entities, which has proven to be a useful prior assumption for downstream QA performance (Lewis et al., 2019) confirmed by our initial experiments. First, we indexed all of the sentences from a Wikipedia dump using the ElasticSearch search engine. We also extract named entities for each sentence in both the Wikipedia corpus and the sentences used as queries. We assume access to a named-entity recognition system, and in this work make use of the spaCy[2] NER pipeline. Then, for a given context-answer pair, we query the index, using the original context sentence as a query, to return a sentence which (1) contains the answer, (2) does not come from the *context*, and (3) has a lower than 95% F1 score with the query sentence to discard highly similar or plagiarized sentences. Besides ensuring that the retrieved sentence and query sentence share the answer entity, we require that at least one additional matching entity appears in both the query sentence and in the entire context, and we perform ablation studies on the effect of this matching below. These retrieved sentences are then fed into our question-generation module.

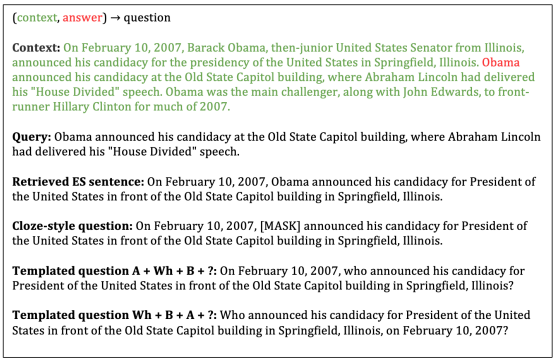

Figure 2: Example of synthetically generated questions using generic cloze-style questions as well as a template-based approach.

Template-based Question Generation: We consider several question styles (1) generic cloze-style questions where the answer is replaced by the token "[MASK]", (2) templated question "Wh+B+A+?" as well as variations on the ordering of this template, as shown in Figure 2. Given the retrieved sentence in the form of `[Fragment A] [Answer] [Fragment B]`, the templated question "Wh+B+A+?" replaces the *answer* with a Wh-component (e.g., what, who, where), which depends on the entity type of the *answer* and places the Wh-component at the beginning of the question, followed by sentence `Fragment B` and `Fragment A`. For the choice of wh-component, we sample a bi-gram based on prior probabilities of that bi-gram being associated with the named-entity type of the answer. This prior probability is calculated based on named-entity and question bi-gram starters from the SQuAD dataset. This information does not make use of the full context-question-answer and can be viewed

[1] https://github.com/awslabs/unsupervised-qa

[2] https://spacy.io

as prior information, not disturbing the integrity of our unsupervised approach. Additionally, the choice of wh component does not significantly affect results. For template-based approaches, we also experimented with clause-based templates but did not find significant differences in performance.

3 Experiments

Settings: For all downstream question answering models, we fine-tune a pretrained BERT model using the Transformers repository (Wolf et al., 2019) and report ablation study numbers using the base-uncased version of BERT, consistent with (Lewis et al., 2019). All models are *trained and validated* on *generated pairs of questions and answers* along with their contexts *tested* on the *SQuAD development set*. The training set differs for each ablation study and will be described below, while the validation dataset is a random set of 1,000 template-based generated data points, which is consistent across all ablation studies. We train all QA models for 2 epochs, checkpointing the models every 500 steps and choosing the checkpoint with the highest F1 score on the validation set as the best model. All ablation studies are averaged over two training runs with different seeds. Unless otherwise stated, experiments are performed using 50,000 synthetic QA training examples, as initial models performed best with this amount. We will make this generated training data public.

3.1 Model Analysis

Effect of retrieved sentences: We test the effect of retrieved vs original sentences as input to question generation when using generic cloze questions. As shown in Table 1, using retrieved sentences improves over using the original sentence, reinforcing our motivation that a retrieved sentence, which may not match trivially the current context, forces the QA model to learn more complex relationships than just simple entity matching. The retrieval process may return sentences which do not match the original context. On a random sample, 15/18 retrieved sentences were judged as entirely relevant to the original sentence. This retrieval is already quite good, as we use a high quality ElasticSearch retrieval and use the original context sentence as the query, not just the answer word. While we do not explicitly ensure that the retrieved sentence has the same meaning, we find that the search results with entity matching gives largely semantically matching sentences. Additionally, we believe the sentences which have loosely related meaning may act as a regularization factor which prevent the downstream QA model from learning only string matching patterns. Along these lines, (Lewis et al., 2019) found that a simple noise function of dropping, masking and permuting words was a strong question generation baseline. We believe that loosely related context sentences can act as a more intuitive noise function, and investigating the role of the semantic match of the retrieved sentences is an important direction for future work. For the sections which follow, we only show results of retrieved sentences, as the trend of improved performance held across all experiments.

Training procedure	EM	F1
Cloze-style original	17.36	25.90
Cloze-style retrieved	**30.53**	**39.61**

Table 1: Effect of original vs retrieved sentences for generic cloze-style question generation.

Effect of template components: We evaluate the effect of individual template components on downstream QA performance. Results are shown in Table 2. Wh template methods improve largely over the simple cloze templates. "Wh + B + A + ?" performs best among the template-based methods, as having the Wh word at the beginning most resembles the target SQuAD domain and switching the order of Fragment B and Fragment A may force the model to learn more complex relationships from the question. We additionally test the effect of the wh-component and the question mark added at the end of the sentence. Using the same data as "Wh + B + A + ?" but removing the wh-component results in a large decrease in performance. We believe that this is because the wh-component signals the type of possible answer entities, which helps narrow down the space of possible answers. Removing the question mark at the end of the template also results in decreased performance, but not as large as removing the wh-component. This may be a result of BERT pretraining which expects certain punctuation based on sentence structure. We note that these questions may not be grammatical, which may have an impact on performance. Improving the question quality makes a difference in performance as seen from the jump from cloze-style questions to template questions. The ablation studies suggest that a combination of question relevance, though

Template data	EM	F1
Cloze	30.53	39.61
A + Wh + B + ?	45.62	55.44
Wh + A + B + ?	44.08	53.90
Wh + B + A + ?	**46.09**	**56.82**
B + A + ?	37.57	46.41
Wh + B + A	44.87	54.56
Wh_simple + B + A + ?	45.60	56.07
What + B + A + ?	10.24	17.04

Table 2: Effect of order of template, wh word and question mark on downstream QA performance.

matching entities, and question formulation, as described above, determine downstream performance. Balancing those two components is an interesting problem and we leave improving grammaticality and fluency through means such as language model generation for future experiments.

In the last two rows of Table 2, we show the effect of using the wh bi-gram prior on downstream QA training. Using the most-common wh word by grouping named entities into 5 categories according to (Lewis et al., 2019) performs very close to the best-performing wh n-gram prior method, while using a single wh-word (what) results in a significant decrease in performance. These results suggest that information about named entity type signaled by the wh-word does provide important information to the model but further information beyond wh-simple does not improve results significantly.

Effect of filtering by entity matching: Besides ensuring that the retrieved sentence and query sentence share the answer entity, we require that at least one additional matching entity appears in both query sentence and entire context. Results are shown in Table 3. Auxillary matching leads to improvements over no matching when using template-based data, with best results using matching with both query and context. Matching may filter some sentences whose topic are too far from the original context. We leave further investigation of the effect of retrieved sentence relevance to future work.

Effect of synthetic training dataset size: Notably, (Lewis et al., 2019) make use of approximately 4 million synthetic data points in order to train their model. However, we are able to train a model with better performance in much fewer examples, and show that such a large subset is unnecessary for their released synthetic training data

Matching procedure	EM	F1
No matching	41.02	50.81
Query matching	44.76	54.87
Context matching	44.22	55.35
Query + Context matching	**46.09**	**56.82**

Table 3: Effect of query and context matching for retrieved input to question generation module on downstream QA performance.

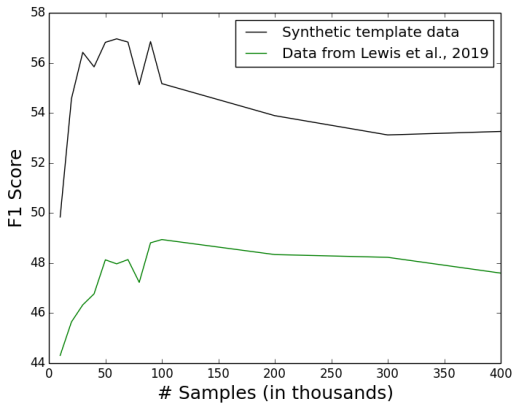

Figure 3: A comparison of the effect of the size of synthetic data on downstream QA performance.

as well. Figure 3 shows the performance from training over random subsets of differing sizes and testing on the SQuAD development data. We sample a random question for each context from the data of (Lewis et al., 2019). Even with as little as 10k datapoints, training from our synthetically generated template-based data with auxiliary matching outperforms the results from ablation studies in (Lewis et al., 2019). Using data from our template-based data consistently outperforms that of (Lewis et al., 2019). Training on either dataset shows similar trends; performance decreases after increasing the number of synthetic examples past 100,000, likely due to a distributional mismatch with the SQuAD data. We chose to use 50,000 examples for our final experiments with other ablation studies as this number gave good performance in initial experiments.

3.2 Comparison of Best-Performing Models:

We compare training on our best template-based data with state-of-the-art in Table 4. SQuAD F1 results reflect results on the hidden SQuAD test set. We report single-model numbers; Lewis et al. (2019) report an ensemble method achieving 56.40 F1 and a best single model achieving 54.7 F1. We make use of the whole-word-masking version of

Model Choice	SQuAD Test F1	SQuAD NER F1
BERT-large (ours)	**64.04**	**77.55**
BERT-large (Lewis et al., 2019)	56.40	64.50

Table 4: A comparison of top results using the BERT-large model.

BERT-large, although using the original BERT-large gives similar performance of 62.69 on the SQuAD dev set. We report numbers on the sample of SQuAD questions which are named entities, which we refer to as SQuAD-NER. The subset corresponding to the SQuAD development dataset has 4,338 samples, and may differ slightly from (Lewis et al., 2019) due to differences in NER preprocessing. We also trained a fully-supervised model on the SQuAD training dataset with varying amounts of data and found our unsupervised performance equals the supervised performance trained on about 3,000 labeled examples.

4 Conclusion

In this paper we introduce a retrieval-based approach to unsupervised extractive question answering. A simple template-based approach achieves state-of-the-art results for unsupervised methods on the SQuAD dataset of 64.04 F1, and 77.55 F1 when the answer is a named entity. We analyze the effect of several components in our template-based approaches through ablation studies. We aim to experiment with other datasets and other domains, incorporate our synthetic data in a semi-supervised setting and test the feasibility of our framework in a multi-lingual setting.

5 Acknowledgements

We thank Xiaofei Ma for fruitful discussions on the project.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and unsupervised transfer learning for question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and effective semi-supervised question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587, New Orleans, Louisiana. 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*, 7:453–466.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. 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*.

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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Liang Wang, Sujian Li, Wei Zhao, Kewei Shen, Meng Sun, Ruoyu Jia, and Jingming Liu. 2018. Multi-perspective context aggregation for semi-supervised cloze-style reading comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 857–867, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtow-

icz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of NeurIPS 2019*.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-supervised QA with generative domain-adaptive nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, Vancouver, Canada. Association for Computational Linguistics.