Developing and Evaluating a Dataset for How-to Tip Machine Reading at Scale

Fuzhu Zhu, Shuting Bai, Tingxuan Li, Takehito Utsuro
Degree Programs in Systems and Information Engineering,
Graduate School of Science and Technology, University of Tsukuba,
1-1-1, Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
{s2220804,s2020817,s2120816}_@_s.tsukuba.ac.jp, utsuro_@_iit.tsukuba.ac.jp

Abstract

In this paper, we focus on the task of machine reading at scale within how-to tip machine reading comprehension (MRC). We propose a method for developing a context dataset using how-to tip websites on the Internet as information sources. This shows that the proposed method can easily create a context dataset containing thousands of context sets. Furthermore, this paper uses a method for retrieving the context from the developed context dataset, which contains the answer of the question. It applies to the MRC model. Specifically, we use three models based on TF-IDF and BERT (TF-IDF, BERT, and TF-IDF+BERT) as our retrieval models. Meanwhile, the BERT model served as our MRC model. We apply the retrieval model and the MRC model to the context dataset after combining them. Evaluation results show that the TF-IDF+BERT model outperforms the other two models when tested against the context dataset.

1 Introduction

In natural language processing, machine reading comprehension (MRC) tasks are formulated to extract the answer to a question from a context within a few question sentences and contexts expressed in natural language. MRC tasks can be divided into two categories based on the two types of answers. Factoid MRC tasks aim at having the answer to factoids such as proper nouns and numbers, where the answer is usually unique, short and simple. Conversely, nonfactoid MRC tasks aim to obtain an answer about nonfactoid such as explanation, reason and how-to tip, where there are usually multiple options and the answer is frequently a full sentence, rahter than a word or phrase. The Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016)is one of the most well-known QA datasets and benchmark tests among factoid MRC related to Wikipedia articles and news articles. Additionally, it is acknowledged that recent deep learning models (for example, BERT (Devlin et al., 2019)) trained with SQuAD achieved fairly high performance[1]. However, some research cases are known for nonfactoid MRC. They include MS MARCO (Nguyen et al., 2016), which has been developed using Bing's search logs and passages of retrieved web pages; DuReader (He et al., 2018), which has been developed using Baidu Search; Baidu Zhidao, a Chinese community-based QA site; and the NarrativeQA (Kočiský et al., 2018) dataset (in English), which contains questions created by editors based on summaries of movie scripts and books. They also include Soleimani et al. (2021), Dulceanu et al. (2018), and Cohen et al. (2018). Among those working on nonfactoid MRC, the case of MRC of Japanese how-to tip QAs (Chen et al., 2020) selected the how-to tip websites that are posted on the Internet and chose the column pages on how-to tip websites as information sources to collect how-to tip QA examples for training and testing. It has also been shown that the how-to tip MRC model with specific performance can be developed.

Figure 1 shows the how-to tip MRC model (Chen et al., 2020). The how-to tip MRC model and the

[1] `https://rajpurkar.github.io/SQuAD-explorer/`

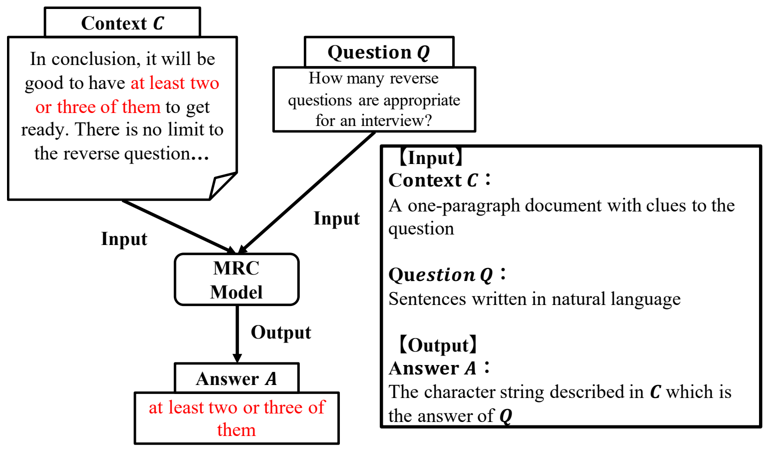


Figure 1: The framework of how-to tip MRC model

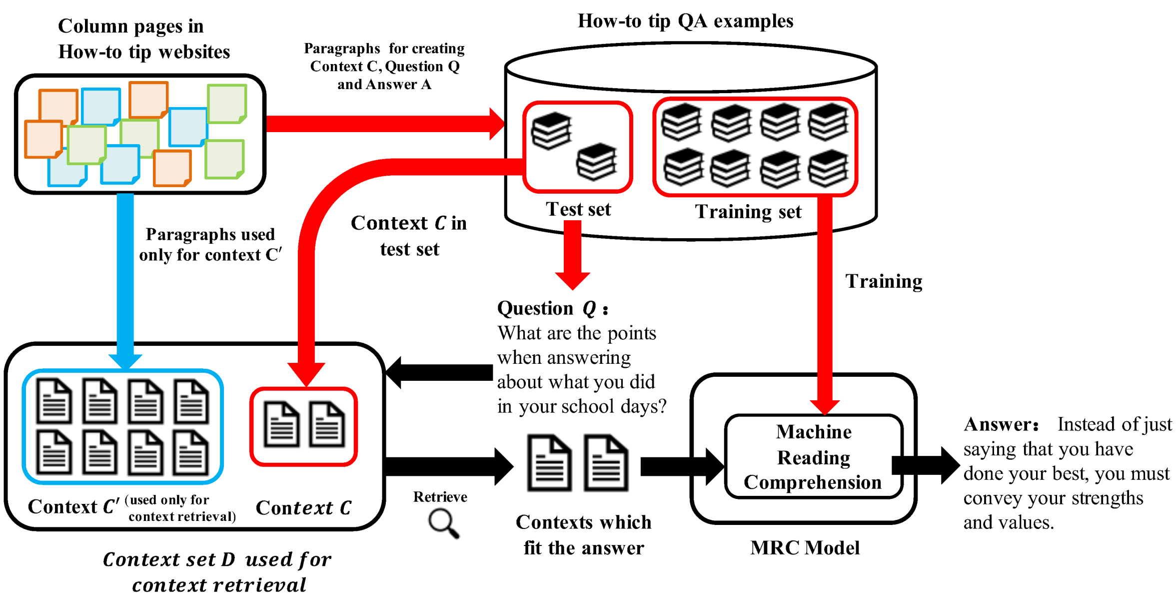


Figure 2: Developing a context dataset for how-to tip MRC by using column pages on how-to tip websites

framework of the typical MRC model, which contains a tuple of a context, a tip question, and an answer, can be represented as in this figure. Note that the answer is extracted only from the context. Therefore, in the situation where it is not given which context to be used, another framework called "machine reading at scale" (Chen et al., 2017) should be invented. In the framework of "machine reading at scale," it handles both information retrieval and MRC tasks. In its framework, the MRC model is applied to the set of candidate contexts retrieved by the information retrieval module. For example, in the information retrieval module, Chen et al. (2017) used the method of TF-IDF to collect the candidate contexts. As another example of "machine reading at scale," using the BERT (Devlin et al., 2019) model as part of the information retrieval model for machine reading at scale tasks has also been studied by Karpukhin et al. (2020). It is shown that using the BERT model as part of the information retrieval model, higher retrieval accuracy than the BM25 method can be achieved in several factoid MRC datasets. Moreover, it shows that the retrieval accuracy was further improved using the proposed retrieval model and the BM25 score together.

Based on that background, this paper applies the framework of "machine reading at scale" to how-to tip MRC. In this paper, we use three different types of retrieval models (context retrieval by TF-IDF (Chen et al., 2017), BERT model, and combining TF-IDF with the BERT model) and how-to tip MRC model (Chen et al., 2020) to how-to tip MRC tasks. Chen et al. (2020) chose the column pages in how-to tip websites as information sources to collect how-to tip QA examples as the training and test sets for the how-to tip MRC model. In this paper, we collect the contexts from the column pages that were not used to form the training and test sets of the how-to tip MRC model in Chen et al. (2020) as shown in the framework in Figure 2 and the example in Figure 3. Then, we use those collected contexts as the contexts C' (used only for context retrieval but not for the MRC model training) for context retrieval and how-to tip MRC task. In this paper, according to the procedures above, we finally develop a dataset for how-to tip machine reading at scale. As for the contexts C', thousands of them are collected.

2 A Dataset for How-to Tip Machine Reading at Scale

In this section, we will introduce how to collect the context C' used only for context retrieval in Figure 2 and how to develop a dataset for how-to tip machine reading at scale.

Japanese how-to tip websites were selected from six types of topics[2] (which are "job hunting," "marriage," "apartment," "hay fever," "dentist," and "food poisoning") by Chen et al. (2020). After that, they collected column pages from the how-to tip websites[3]. Finally, a maximum of five paragraphs were selected from each column page, and they used them as contexts for constructing answerable/unanswerable how-to tip QA examples. An answerable how-to tip QA example contains Context C, Question Q, and Answer A, whereas an unanswerable how-to tip QA example contains Context C, Question Q, and Answer $A' = \langle \text{null} \rangle$[4].

Considering the above procedure of Chen et al. (2020), this section shows how we collect the context C' used only for context retrieval in Figure 2. More specifically, as shown in the example of Figure 3, within the column page used by Chen et al. (2020), we do not use the maximum five paragraphs selected by Chen et al. (2020) (as shown in the red boxes). Still, we use those other than the maximum five paragraphs (as shown in the blue boxes). We also carefully examine the context dataset of Chen et al. (2020), which was developed manually by selecting the paragraph used, and we follow the standards below to select the candidate paragraphs efficiently:[5,6]

(i) Based on the restriction when applying the MRC models by BERT (Devlin et al., 2019), the upper bound of the number of morphemes within a paragraph is set to 290[7].
(ii) The lower bound of the number of characters in a paragraph is 30.
(iii) Any URL is excluded from the paragraph.
(iv) Any email addresses were excluded from the paragraph.

Table 1 shows the number of web pages used for each topic ("job hunting," "marriage," "apartment," "hay fever," "dentist," and "food poisoning"). It also shows the number of contexts used for constructing how-to tip QA examples and the number of contexts used only for context retrieval[8]. Figure 3 shows how

[2]The specific term used in Chen et al. (2020) is "query focus," rather than "topic." The notion of *query focus* is a keyword used for every search request related to a specific subject. In this paper, however, for simplicity, we use the term "topic" in stead of "query focus."

[3]The detailed procedure of collecting pages from the how-to tip websites is stated by Chen et al. (2020).

[4]Both SQuAD1.1-type answerable and SQuAD2.0-type unanswerable QA examples were created from the same column page (Chen et al., 2020).

[5]The standard (i) is simply for satisfying the requirement when applying the MRC models by BERT. Conversely, the standards (ii), (iii), and (iv) are for avoiding paragraphs that do not have sufficient how-to tip knowledge. These standards are also for avoiding the task of filtering out the context C' used only for the context retrieval to be too easy.

[6]One of the authors of the paper performed the procedure of manual selection.

[7]In the experiments and evaluation of the retrieval module throughout this paper, MeCab (`https://taku910.github.io/mecab/`) and mecab-ipadic-NEologd (`https://github.com/ neologd/mecab-ipadic-neologd`) are used in Japanese morphological analysis.

[8]Those data shown in Table 1 (except for "The number of contexts only for retrieval"), Table 2, and Table 3 are essentially the same as those reported in Chen et al. (2020), but we

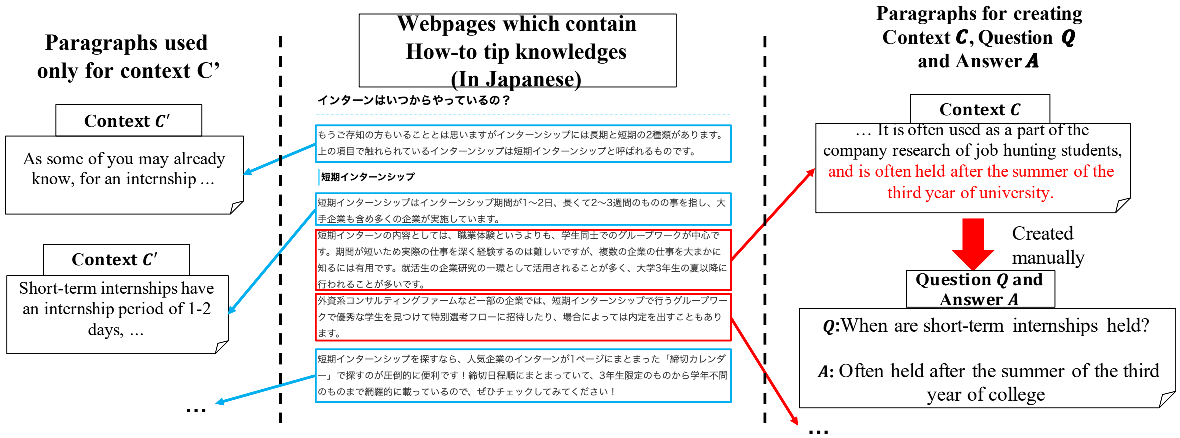


Figure 3: Using a column page to collect contexts for creating how-to tip QA examples (from: "When does job hunting begin?" (in Japanese) (https://internshipguide.jp/columns/view/shukatsu_sched_1))

Table 1: The number of used web pages and the number of collected contexts

Topic	Number of used web pages	The number of contexts for QA examples		The number of contexts only for retrieval
		Training examples	Test examples	
job hunting	293	1,478	98	4,675
marriage	182	1,386	98	2,868
apartment	50	—	100	491
hay fever dentist food poisoning	51	—	100	962

to collect contexts from a column page. Based on the procedures above, as shown in Figure 2, the context set D used for context retrieval consists of the context set C' used only for context retrieval and the context set C of the test examples of how-to tip QA examples.

3 BERT Retrieval Model

This section describes the structure of the BERT retrieval model devdeloped based on Karpukhin et al. (2020), the training method, and the retrieval procedure.

This BERT retrieval model uses two independent BERT models (Devlin et al., 2019)[9] as a question encoder E_q and a context encoder (in Karpukhin et al. (2020), passage encoder) E_c. The BERT model is applied to each input question Q and context C and the representations of the output CLS tokens are used as the representations $E_q(Q)$ and $E_c(C)$ of question Q and context C. The cosine similarity of the following equation is used as the similarity between the encoded Q and C.

$$sim(Q, C) = \frac{E_q(Q) \cdot E_c(C)}{\parallel E_q(Q) \parallel \ \parallel E_c(C) \parallel} \quad (1)$$

In the process of training the model, for $i = 1, \ldots, m$, a set of the question Q_i, one relevant (positive) context C_i^+ that contains the reference answer and n irrelevant (negative) contexts C_i^- that do not contain the reference answer, is used as a training instance.

$$(Q_i, C_i^+, C_{i,1}^-, \ldots, C_{i,n}^-) \quad (2)$$

and m sets of such a tuple are collected as a set T of training data.

$$T = \left\{ (Q_i, C_i^+, C_{i,1}^-, \ldots, C_{i,n}^-) \middle| i = 1, \ldots, m \right\}$$

We optimize the loss function below that is the neg-

increase their numbers through annotation to additional data. The reason why the number of examples for "apartment," "hay fever," "dentist," and "food poisoning" is less than those of "job hunting" and "marriage" is simply that the annotation procedure had started from "job hunting" and "marriage." It is quite possible to collect the same number of examples for each topic "apartment," "hay fever," "dentist," and "food poisoning."

[9] The BERT retrieval model was implemented using the HuggingFace version (https://github.com/huggingface/ transformers). A multilingual cased model was adopted as the pre-training model.

ative log likelihood of the positive context:

$$L(Q_i, C_i^+, C_{i,1}^-, \ldots, C_{i,n}^-) = - log \frac{e^{sim(Q_i, C_i^+)}}{e^{sim(Q_i, C_i^+)} + \sum_{j=1}^{n} (e^{sim(Q_i, C_{i,j}^-)})} \quad (3)$$

Furthermore, to create the training dataset of a question Q and a context C that contains the reference answer above, we follow the strategy of "in-batch negatives" of Karpukhin et al. (2020). In this strategy, assume that we have B questions in a mini-batch and each one is associated with a relevant context. Roughly speaking, for each question Q_i in a mini-batch, there exist $B-1$ contexts, each of which is the relevant context of one of other $B-1$ questions in the same mini-batch. However, for the question Q_i, each of those $B-1$ contexts can be regarded as an irrelevant context. With this strategy, it enables us to create B training instances in each batch, where there are $B-1$ negative contexts for each question. This strategy is known as effective for boosting the number of training examples.

When we retrieve the contexts, the fine-tuned BERT model is used to pre-encode the contexts used for context retrieval, where the contexts are indexed using FAISS (Johnson et al., 2021) offline. For each question, the Top n contexts are output as retrieval results under the similarity scale of the formula (1).

4 Evaluation

4.1 The Dataset

Table 1 shows the number of web pages and the number of contexts used for creating how-to tip QA examples, as well as the number of contexts used only for context retrieval in the evaluation. Table 2 also shows the number of questions in how-to tip QA examples and Table 3 shows the number of how-to tip QA examples and factoid QA examples, respectively.

4.2 Evaluation Procedure

We use the following three types of context retrieval models to evaluate our dataset.

(i) TF-IDF model.

(ii) BERT retrieval model.

Table 2: Number of questions related to how-to tip

topic	For creating Training set	For creating Test set
job hunting	795	50
marriage	799	49
apartment	—	50
others	—	49

Table 3: The number of QA examples

(a) factoid QA examples

training/test	The number of sets of context，question and answer (answerable/unanswerable)
training	$27,427/28,742$
test	$50/50$

(b) how-to tip QA examples

topic	The number of sets of context, question and answer (answerable/unanswerable)	
	Training set	Test set
job hunting	$807/807$	$50/50$
marriage	$807/807$	$50/50$
apartment	—	$50/50$
others	—	$50/50$

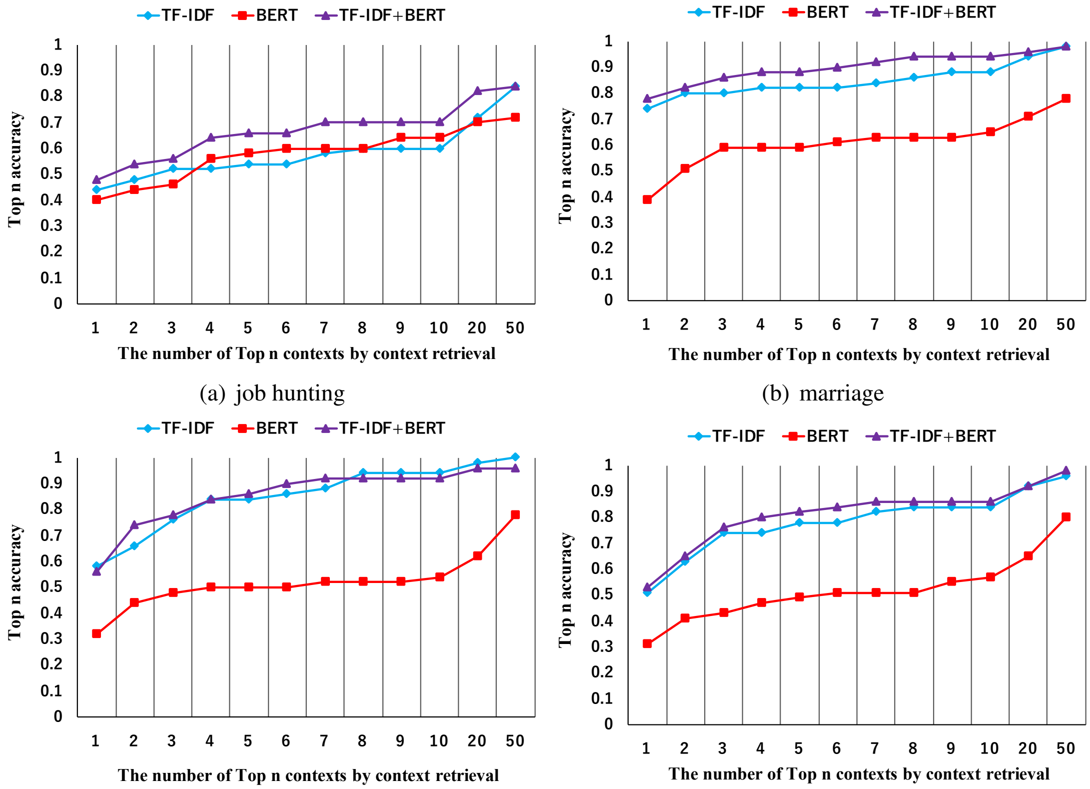
(iii) "TF-IDF+BERT" model, which takes the sum of (i) and (ii) scores.

For (i), to build the TF-IDF (Chen et al., 2017) model[10], we add a stop word list in Japanese-SlothLib[11]. For each context set of the topics of "job hunting," "marriage," "apartment," and the mixture of "hay fever," "dentist," and "food poisoning," one TF-IDF model is built.

As described in Section 3, for (ii), we use the set of the pairs of question Q and the context C that contains the reference answer as the training data. The numbers of the set of the question Q, the context C, and the answer A are as shown in Table 3(b), where we use only the answerable training data for the topics "job hunting" and "marriage" and fine-tune the BERT retrieval model.

[10] `https://github.com/facebookresearch/DrQA`

[11] `http://svn.sourceforge.jp/svnroot/slothlib/`



(a) job hunting

(b) marriage

(c) apartment

(d) hay fever / dentist / food poisoning

Figure 4: Evaluation results of the three types of context retrieval models (with top n accuracy of the retrieved contexts)

For (iii), we use the inner product of the TF-IDF models feature vector of the question Q and the context C as the score $S_T(Q, C)$ of the TF-IDF model and use the cosine similarity between Q and C that are encoded by the BERT retrieval model as the score $S_B(Q, C)$. For one question Q_i, suppose that $S_T(Q_i, C_j)(j = 1, \ldots, n)$ are the scores for the n candidate contexts[12]; the following equation gives the score $S_{T+B}(Q_i, C_j)$ of the "TF-IDF+BERT" model, which is the sum of the scores of (i) and (ii):

$$S_{T+B}(Q_i, C_j) = S_T(Q_i, C_j) + S_B(Q_i, C_j) \quad (4)$$

Based on S_{T+B}, we rank the candidate contexts, and the top k ($k = 1, \ldots, n$) contexts are output as the results.

Meanwhile, the following three types of QA examples are used to fine-tune the BERT (Devlin et al., 2019) MRC model.

(i) Factoid QA examples (the training examples are shown in Table 3(a)).

(ii) How-to tip QA examples of "job hunting" and "marriage" (the training examples are shown in Table 3(b)).

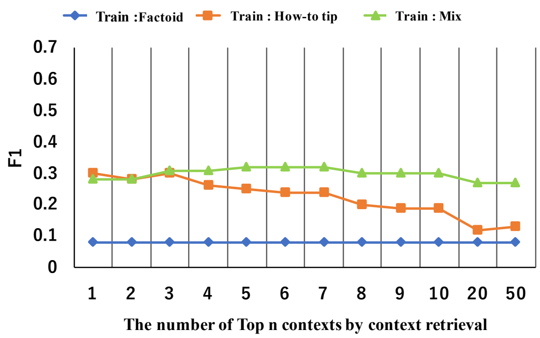
(iii) A mixture of both (i) and (ii).

As the version of the BERT implementation, which can handle a text in Japanese, the TensorFlow version[13] and the Multilingual Cased model[14] were used as the pre-trained model. Before applying BERT modules, MeCab was applied with IPAdic dictionary, and the Japanese text was segmented into a morpheme sequence. Then, within the BERT fine-tuning module, the WordPiece module with 110k shared WordPiece vocabulary was applied, and the Japanese text was further segmented into a subword

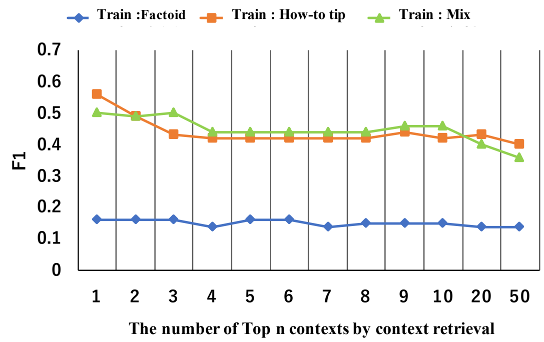
[12]The score $S_T(Q_i, C_j)(j = 1, \ldots, n)$ is supposed to be normalized by the Min-Max method (minimum value is 0, whereas the maximum value is 1).

[13]`https://github.com/google-research/bert`

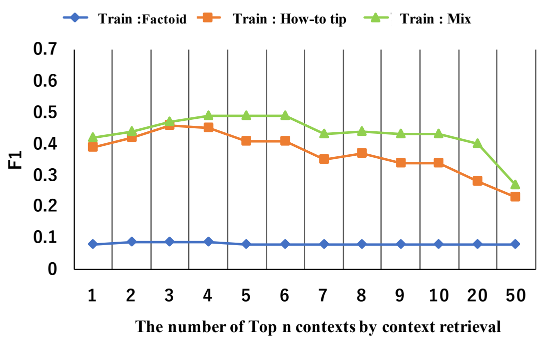
[14]Trained in 104 languages, available from `https://github.com/google-research/bert/blob/master/multilingual.md`.



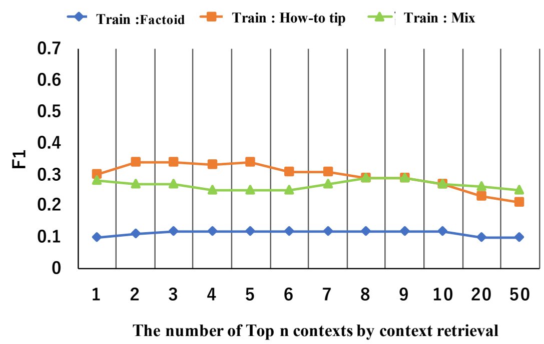
(a) job hunting



(b) marriage

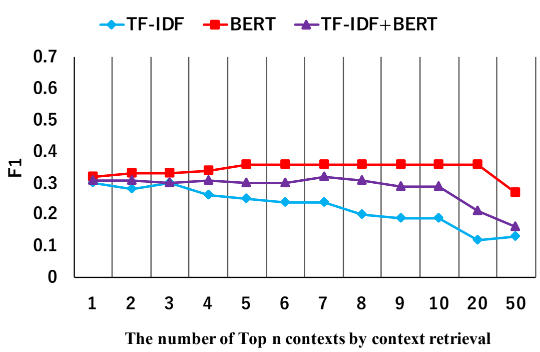


(c) apartment

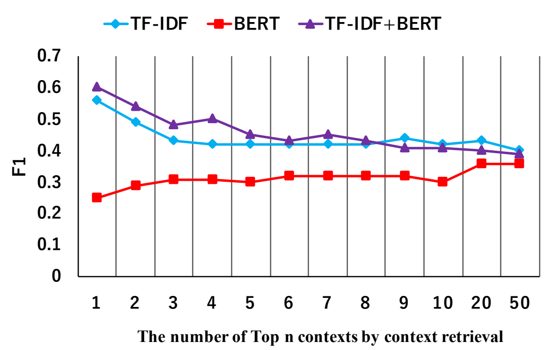


(d) hay fever / dentist / food poisoning

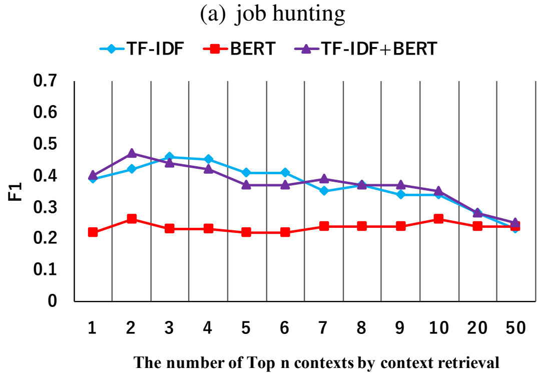
Figure 5: Evaluation results of machine reading at scale for three types of datasets used to fine-tune the BERT MRC model (with the TF-IDF model for context retrieval)



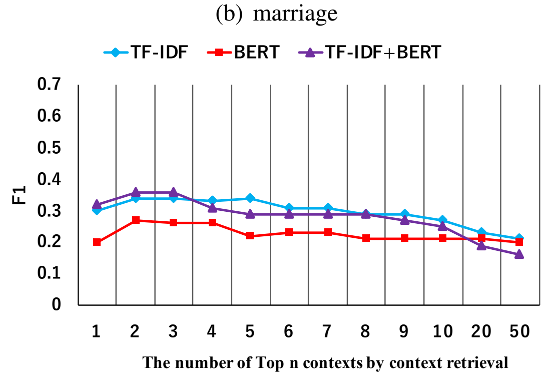
(a) job hunting



(b) marriage

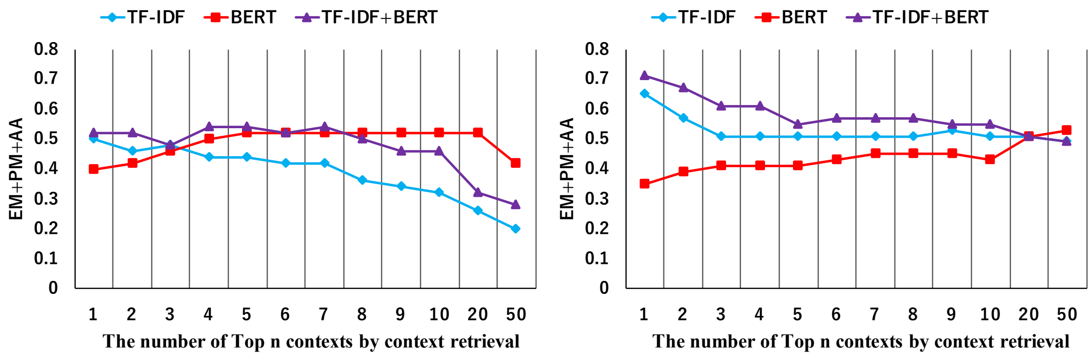


(c) apartment



(d) hay fever / dentist / food poisoning

Figure 6: Evaluation results of the three types of context retrieval models (with the MRC model trained with how-to tip QA examples)



(a) job hunting (b) marriage

Figure 7: Manual evaluation results of the three types of context retrieval models (with the MRC model trained with how-to tip QA examples)

unit sequence. Finally, the BERT fine-tuning module for MRC model[15] was applied.

The how-to tip MRC model is applied to top n ($n = 1, \dots, 50$) retrieved contexts. Then, the answer with the highest score of the MRC model is chosen as the MRC model's output. Finally, we measure the F1 score which is calculated against the morpheme sequence of the reference answer.

In the manual evaluation[16], comparing the model's output and the reference answer, we evaluate the result manually according to the three evaluation criteria of "Exact Match" (EM), "Partial Match" (PM) and "Another Answer" (AA). We consider it the criterion for "Partial Match," when sufficient but partial information overlaps between the model's output and the reference answer. The criterion on "Another Answer," we consider it an answer when the condition "It is different from the reference answer, but contains enough information to answer the question" is satisfied. Then, we can calculate the ratio of the numbers of "Exact Match", "Partial Match" and "Another Answer" (EM+PM+AA).

4.3 Evaluation Result

Figure 4 shows the results of evaluating three types of context retrieval models in terms of top n retrieval accuracy, measured as the rate of queries for which the top n contexts include those with the reference answers. Figure 4(a) shows that the BERT retrieval model performs worse for cases other than "job hunting." This is mainly because, for the topics other than "job hunting," the queries for evaluation tend to include morphemes that appear in the contexts with the reference answers, which makes the TF-IDF model perform much better than the BERT retrieval model. For topic "job hunting," however, the queries for evaluation tend to include a relatively small number of morphemes that appear in the contexts with the reference answers, which happens to benefit the BERT retrieval model and makes it perform comparatively well with the TF-IDF model. By simply adding the scores of the two models, the "TF-IDF+BERT" model performs the best.

Figure 5 compares the three types of datasets used to fine-tune the BERT MRC model where the TF-IDF model is used for context retrieval. Similar to the evaluation results in Chen et al. (2020), also in the case of how-to tip MRC at scale in this paper, the performance of the model trained only by the factoid QA examples was significantly worse, whereas the one trained with the mixture of factoid + how-to tip QA examples performed the best. Overall, as the number of top n contexts increases, the model's performance tends to decrease on the contrary. This is simply because, as the number of top n contexts increases, not only those contexts with the reference answer, but also other contexts are included in the top n contexts, which damages the final MRC model results.

Figure 6 also compares the three types of context retrieval models, where the MRC model is trained with how-to tip QA examples[17]. Similarly, in Figure 4, the TF-IDF model performs well. Also, from

[15] `run_squad.py`, with the number of epochs of 2, batch size of 8, and learning rate of 0.00003.

[16] One of the author of the paper conducted a manual evaluation.

[17] The MRC model trained with the mixture of factoid + how-to tip QA examples shows almost a similar performance.

both Figure 5 and Figure 6, it can be seen that the MRC model trained with the topics of "job hunting" and "marriage" performs fairly well in how-to tip MRC on other topics such as "apartment," "hay fever," "dentist," and "food poisoning." From this result, it is sufficient to collect how-to tip QA examples only for one or two topics such as 'job hunting" and "marriage," and then fine-tune the MRC model, which applies to how-to tip MRC of any topic.

Finally, Figure 7 shows the manual evaluation result of the MRC model trained with how-to tip QA examples. Overall, the "TF-IDF+BERT" model performs the best in the evaluation of the performance of the MRC model for the topics of "job hunting" and "marriage." Compared with the automatic F1 results in Figure 6, it seems that the relative performance of the "TF-IDF+BERT" model improves simply because, by manual evaluation, certain nonliteral expressions within the "Another Answer" contribute to improving the performance of the "TF-IDF+BERT" model.

5 Related Work

Related studies of machine reading at scale, i.e., Chen et al. (2017), Karpukhin et al. (2020), Nishida et al. (2018), and Lee et al. (2019) investigated machine reading at scale in the context of factoid MRC. In Chen et al. (2017), machine reading at scale is realized by combining TF-IDF, which is used to realize context retrieval, and a neural MRC model using RNN. Karpukhin et al. (2020) used BERT (Devlin et al., 2019) for retrieval and then applied it to build a system for machine reading at scale. Moreover, in Nishida et al. (2018), machine reading at scale is realized via multi-task learning of information retrieval and MRC. Meanwhile, Lee et al. (2019) proposed an end-to-end framework for machine reading at scale that trains the retrieval and reading comprehension models.

In this paper, similar to Chen et al. (2017), TF-IDF model is used for the context retrieval part compared with those previous works, whereas another retrieval model (Karpukhin et al., 2020) by BERT is also investigated in this paper. For the part of reading comprehension, we use the BERT model instead. Combining these two parts, machine reading at scale is realized. Compared with that of Karpukhin et al. (2020), it is also important to note that we evaluate the performance change of the MRC model when the number of top n contexts increases, where it is observed that, in the case of our how-to tip QA examples, the optimal performance is around $n = 1$.

6 Conclusion

In this paper, we proposed a method to collect the contexts from the column pages that are not used to train the MRC model in Chen et al. (2020) and then use them to evaluate how-to tip machine reading at scale. Then, consequently, we developed a dataset that contains thousands of contexts for how-to tip machine reading at scale. Furthermore, we evaluated the three types of context retrieval models and showed that the "TF-IDF+BERT" model is the most effective. Future works include expanding the dataset as well as designing the evaluation procedure to be more reliable by introducing the notion of repeated trials and considering statistical measures such as variance (Dodge et al., 2020).

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 21H00901.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July. Association for Computational Linguistics.

Tengyang Chen, Hongyu Li, Miho Kasamatsu, Takehito Utsuro, and Yasuhide Kawada. 2020. Developing a how-to tip machine comprehension dataset and its evaluation in machine comprehension by BERT. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 26–35, Online, July. Association for Computational Linguistics.

Daniel Cohen, Liu Yang, and W. Bruce Croft. 2018. WikiPassageQA: A benchmark collection for research on non-factoid answer passage retrieval. In *Proceedings of the 41th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018)*, pages 1165–1168, Ann Arbor, Michigan, U.S.A.

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, June. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *Computing Research Repository, arXiv:2002.06305*.

Andrei Dulceanu, Thang Le Dinh, Walter Chang, Trung Bui, Doo Soon Kim, Manh Chien Vu, and Seokhwan Kim. 2018. PhotoshopQuiA: A corpus of non-factoid questions and answers for why-question answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia, July. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July. Association for Computational Linguistics.

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. *Computing Research Repository*, arXiv:1611.09268.

Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 963–966.

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, November. Association for Computational Linguistics.

Amir Soleimani, Christof Monz, and Marcel Worring. 2021. NLQuAD: A non-factoid long question answering data set. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1245–1255, Online, April. Association for Computational Linguistics.