Adaptive Document Retrieval for Deep Question Answering

Bernhard Kratzwald and **Stefan Feuerriegel**
Chair of Management Information Systems
ETH Zurich
Zurich, Switzerland
{bkratzwald, sfeuerriegel}@ethz.ch

Abstract

State-of-the-art systems in deep question answering proceed as follows: (1) an initial document retrieval selects relevant documents, which (2) are then processed by a neural network in order to extract the final answer. Yet the exact interplay between both components is poorly understood, especially concerning the number of candidate documents that should be retrieved. We show that choosing a static number of documents – as used in prior research – suffers from a noise-information trade-off and yields suboptimal results. As a remedy, we propose an adaptive document retrieval model. This learns the optimal candidate number for document retrieval, conditional on the size of the corpus and the query. We report extensive experimental results showing that our adaptive approach outperforms state-of-the-art methods on multiple benchmark datasets, as well as in the context of corpora with variable sizes.

1 Introduction

Question-answering (QA) systems proceed by following a two-staged process (Belkin, 1993): in a first step, a module for document retrieval selects n potentially relevant documents from a given corpus. Subsequently, a machine comprehension module extracts the final answer from the previously-selected documents. The latter step often involves hand-written rules or machine learning classifiers (c. f. Shen and Klakow, 2006; Kaisser and Becker, 2004), and recently also deep neural networks (e. g. Chen et al., 2017; Wang et al., 2018)

The number of candidate documents n affects the interplay between both document retrieval and machine comprehension component. A larger n improves the recall of document retrieval and thus the chance of including the relevant information. However, this also increases the noise and might adversely reduce the accuracy of answer extraction. It was recently shown that a top-1 system can potentially outperform a system selecting more than one document (Kratzwald and Feuerriegel, 2018). This finding suggests that a static choice of n can result a suboptimal performance.

Contributions. This work analyzes the interplay between document retrieval and machine comprehension inside neural QA systems. We first reason numerically why a fixed choice of n in document retrieval can negatively affect the performance of question answering. We thus propose a novel machine learning model that adaptively selects the optimal n_i for each document retrieval. The resulting system outperforms state-of-the-art neural question answering on multiple benchmark datasets. Notably, the overall size of the corpus affects the optimal n considerably and, as a result, our system evinces as especially superior over a fixed n in settings where the corpus size is unknown or grows dynamically.

2 Related Work

Taxonomy of QA systems. Question answering systems are frequently categorized into two main paradigms. On the one hand, knowledge-based systems draw upon manual rules, ontologies and large-scale knowledge graphs in order to deduce answers (e. g. Berant et al., 2013; Lopez et al., 2007; Unger et al., 2012). On the other hand, QA system incorporate a document retrieval module which selects candidate documents based on a chosen similarity metric, while a subsequent module then processes these in order to extract the answer (e. g. Cao et al., 2011; Harabagiu et al., 2000).

Deep QA. Recently, Chen et al. (2017) developed a state-of-the-art deep QA system, where the

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 576–581
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

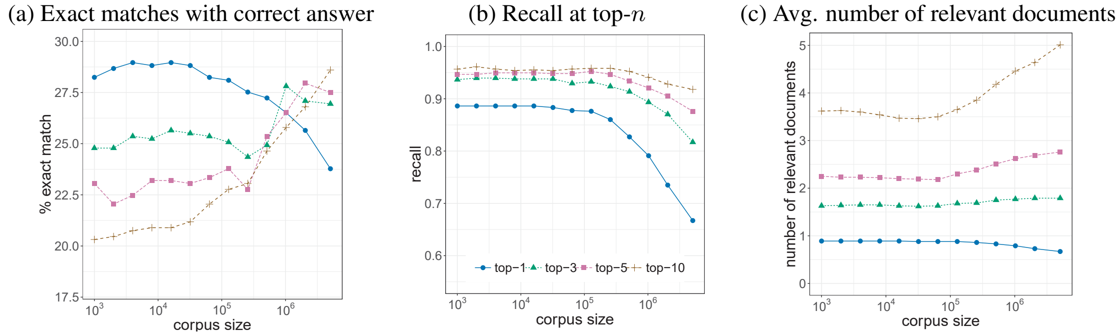


Figure 1: Comparison of how top-n document retrieval affects deep QA. Plot (a) shows the percentage of exact matches with the correct answering, thereby measuring the end-to-end performance of the complete system. Plot (b) gives the recall at top-n, i. e. the fraction of samples where *at least once* the correct answer is returned. Plot (c) depicts the *average number* of documents that contain the ground-truth answer. As a result, the recall lowers with increasing corpus size, yet this not necessarily compromises a top-n system, as it often contains the correct answer more than once.

answer is extracted from the top $n = 5$ documents. This choice stems from computing the dot product between documents and a query vector; with tf-idf weighting of hashed bi-gram counts. Wang et al. (2018) extended this approach by implementing a neural re-ranking of the candidate document, yet keeping the fixed number of n selected documents unchanged. In particular, the interplay between both modules for document retrieval and machine comprehension has not yet been studied. This especially pertains to the number of candidate documents, n, that should be selected during document retrieval.

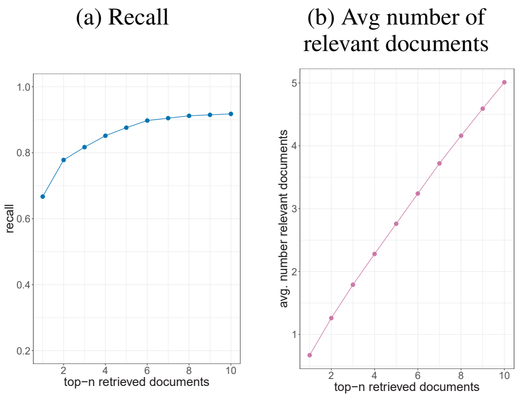
Component interactions. Extensive research has analyzed the interplay of both document retrieval and machine comprehension in the context of knowledge-based systems (c. f. Moldovan et al., 2003) and even retrieval-based systems with machine learning (c. f. Brill et al., 2002). However, these findings do not translate to machine comprehension with deep learning. Deep neural networks consist of a complex attention mechanism for selecting the context-specific answer (Hermann et al., 2015) that has not been available to traditional machine learning and, moreover, deep learning is highly sensitive to settings involving multiple input paragraphs, often struggling with selecting the correct answer (Clark and Gardner, 2017).

3 Noise-Information Trade-Off in Document Retrieval

In the following, we provide empirical evidence why a one-fits-all n can be suboptimal. For this purpose, we run a series of experiments in order to obtain a better understanding of the interplay between document retrieval and machine comprehension modules. That is, we specifically compare the recall of document retrieval to the end-to-end performance of the complete QA system; see Fig. 1. Our experiments study the sensitivity along two dimensions: on the one hand, we change the number of top-n documents that are returned during document retrieval and, on the other hand, we vary the corpus size.

Our experiments utilize the TREC QA dataset as a well-established benchmark for open-domain question answering. It contains 694 question-answer pairs that are answered with the help of Wikipedia. We vary the corpus between a small case (where each question-answer pair contains only one Wikipedia article with the correct answer plus 50 % articles as noise) and the complete Wikipedia dump containing more than five million documents. Our experiments further draw upon the DrQA system (Chen et al., 2017) for question answering that currently stands as a baseline in deep question answering. We further modified it to return different numbers of candidate documents.

Fig. 1 (a) shows the end-to-end performance across different top-n document retrievals as measured by the exact matches with ground truth. For a small corpus, we clearly register a superior performance for the top-1 system. However, we observe a different pattern with increasing corpus size. Fig. 1 (b) and (c) shed light into the underlying reason by reporting how frequently the correct answer is returned and, as the correct an-


(a) Recall

(b) Avg number of relevant documents

Figure 2: Recall (a) and average number of relevant documents (b) for growing top-n configurations and a static corpus size (full Wikipedia dump). While the recall is converging the number of relevant documents keeps growing resulting in a higher density of relevant information.

swer might appear multiple times, how often it is included in the top-n. Evidently, the recall in (b) drops quickly for a top-1 system when augmenting the corpus. Yet it remains fairly stable for a top-n system, due to the fact that it is sufficient to have the correct answer in any of the n documents. According to (c), the correct answer is often more than once returned by a top-n system, increasing the chance of answer extraction.

The above findings result in a noise-information trade-off. A top-1 system often identifies the correct answer for a small corpus, whereas a larger corpus introduces additional noise and thus impedes the overall performance. Conversely, a top-n system accomplishes a higher density of relevant information for a large corpus as the answer is often contained multiple times. This effect is visualized in an additional experiment shown in Fig. 2. We keep the corpus size fixed and vary only n, i.e. the number of retrieved documents. We see the recall converging fast, while the average number of relevant documents keeps growing, leading to a higher density of relevant information. As a result, a top-n system might not be compromised by a declining recall, since it contains the correct answer over-proportionally often. This logic motivates us in the following to introduce an adaptive n_i that optimizes the number of documents retrievals in a top-n system independently for every query q_i.

4 Adaptive Document Retrieval

This section advances deep question answering by developing adaptive methods for document retrieval. Our methods differ from conventional document retrieval in which the number of returned documents is set to a fixed n. Conversely, we actively optimize the choice of n_i for each document retrieval i. Formally, we select n_i between 1 and a maximum τ (e. g. $\tau = 20$), given documents $[d_i^{(1)}, \ldots, d_i^{(\tau)}]$. These entail further scores denoting the relevance, i. e. $s_i = [s_i^{(1)}, \ldots, s_i^{(\tau)}]^T$ with normalization s. t. $\sum_j s_i^{(j)} = 1$. The scoring function is treated as a black-box and thus can be based on simple tf-idf similarity but also complex probabilistic models.

4.1 Threshold-Based Retrieval

As a naïve baseline, we propose a simple threshold-based heuristic. That is, n_i is determined such that the cumulative confidence score reaches a fixed threshold $\theta \in (0, 1]$. Formally, the number n_i of retrieved documents is given by

$$n_i = \max_k \sum_{j=1}^{k} s_i^{(j)} < \theta. \tag{1}$$

In other words, the heuristic fills up documents until surpassing a certain confidence threshold. For instance, if the document retrieval is certain that the correct answer must be located within a specific document, it automatically selects fewer documents.

4.2 Ordinal Regression

We further implement a trainable classifier in the form of an ordinal ridge regression which is tailored to ranking tasks. We further expect the cumulative confidence likely to be linear. The classifier then approximates n_i with a prediction y_i that denotes the position of the first relevant document containing the desired answer. As such, we learn a function

$$y_i = f([s_i^{(1)}, \ldots, s_i^{(\tau)}]) = \lceil s_i^T \beta \rceil, \tag{2}$$

where $\lceil \ldots \rceil$ denotes the ceiling function.

The ridge coefficients are learned through a custom loss function

$$\mathcal{L} = \left\| \lceil X\beta \rceil - y \right\|_1 + \lambda \left\| \beta \right\|_2, \tag{3}$$

where X is a matrix containing scores of our training samples. In contrast to the classical ridge regression, we introduce a ceiling function and replace the mean squared error by a mean absolute error in order to penalize the difference from the optimal rank. The predicted cut-off $\hat{n}_i$ for document retrieval is then computed for new observations ${s'}_i$ via $\hat{n}_i = \lceil {s'}_i^T \hat{\beta} \rceil + b$. The linear offset b is added in order to ensures that $n_i \leq \hat{n}_i$ holds, i. e. reducing the risk that the first relevant document is not included.

We additionally experimented with non-linear predictors, including random forests and feed-forward neural networks; however; we found no significant improvement that justified the additional model complexity over the linear relationship.

5 Experiments

We first compare our QA system with adaptive document retrieval against benchmarks from the literature. Second, we specifically study the sensitivity of our adaptive approach to variations in the corpus size. All our experiments draw upon the DrQA implementation (Chen et al., 2017), a state-of-the-art system for question answering in which we replaced the default module for document retrieval with our adaptive scheme (but leaving all remaining components unchanged, specifically without altering the document scoring or answer extraction).

For the threshold-based model, we set $\tau = 15$ and the confidence threshold to $\theta = 0.75$. For the ordinal regression approach, we choose $\tau = 20$ and use the original SQuAD train-dev split from the full corpus also as the basis for training across all experiments.

5.1 Overall Performance

In a first series of experiments, we refer to an extensive set of prevalent benchmarks for evaluating QA systems, namely, SQuAD (Rajpurkar et al., 2016), Curated TREC (Baudiš and Šedivý, 2015), WikiMovies (Miller et al., 2016) and WebQuestions (Berant et al., 2013) in order to validate the robustness of our findings. Based on these, we then evaluate our adaptive QA systems against the naïve DrQA system in order to evaluate the relative performance. We included the deep QA system R^3 as an additional, top-scoring benchmark from recent literature (Wang et al., 2018) for better comparability.

Tbl. 1 reports the ratio of exact matches for the different QA systems. The results demonstrate the effectiveness of our adaptive scheme: it yields the best-performing system for three out of four datasets. On top of that, it outperforms the naïve DrQA system consistently across all datasets.

5.2 Sensitivity: Adaptive QA to Corpus Size

We earlier observed that the corpus size affects the best choice of n and we thus study the sensitivity with regard to the size. For this purpose, we repeat the experiments from Section 3 in order to evaluate the performance gain from our adaptive scheme. More precisely, we compare the ordinal regression ($b = 1$) against document retrieval with a fixed document count n.

Fig. 3 shows the end-to-end performance, confirming the overall superiority of our adaptive document retrieval. For instance, the top-1 system reaches a slightly higher rate of exact matches for small corpus sizes, but is ranked last when considering the complete corpus. The high performance of the top-1 system partially originates from the design of the experiment itself, where we initially added *one* correct document per question, which is easy to dissect by adding little additional noise. On the other hand, the top-10 system accomplishes the best performance on the complete corpus, whereas it fails to obtain an acceptable performance for smaller corpus sizes.

To quantify our observations, we use a notation of regret. Formally, let μ_{nm} denote the performance of the top-n system on a corpus of size m. Then the regret of choosing system n at evaluation point m is the difference between the best performing system μ_m^* and the chosen system $r_{nm} = \mu_m^* - \mu_{nm}$. The total regret of system n is computed by averaging the regret over all observations of system n, weighted with the span in-between observations in order to account for the logarithmic intervals. The best top-n system yields a regret of 0.83 and 1.12 respectively, whereas our adaptive control improves it down to 0.70.

5.3 Robustness Check

Experiments so far have been conducted on the DrQA system. To show the robustness of our approach, we repeat all experiments on a different QA system. Different from DrQA, this system operates on paragraph-level information retrieval and

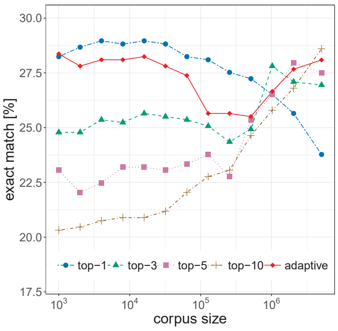


Figure 3: End-to-end performance of adaptive information retrieval over static top-n configurations and a growing corpus.

	SQuAD	TREC	WebQuestions	WikiMovies
DrQA (Chen et al., 2017)†	29.3	27.5	18.5	36.6
Threshold-based ($\theta = 0.75$)	**29.8**	28.7	19.2	38.6
Ordinal regression ($b = 1$)	29.7	28.1	19.4	38.0
Ordinal regression ($b = 3$)	29.6	**29.3**	**19.6**	38.4
R^3 (Wang et al., 2018)	29.1	28.4	17.1	**38.8**

†: Numbers vary slightly from those reported in the original paper, as the public repository was optimized for runtime performance.

Table 1: End-to-end performance of the plain DrQA system measured in exact matches. Performance of two threshold based and two regression based adaptive retreival improvements as well as other state-of-the art systems. Experiments are based on the full Wikipedia dump containing more than 5 million documents.

	SQuAD	TREC	WebQuestions	WikiMovies
Top-50 System	27.0	23.5	15.1	24.4
Top-80 System	27.2	25.9	14.9	26.0
Threshold-based ($\theta = 0.75, \tau = 100$)	27.2	**27.1**	15.4	26.3
Ordinal regression ($b = 3, \tau = 250$)	**27.3**	**27.1**	**16.7**	**26.5**

Table 2: End-to-end performance measured in percentages of exact matching answers of a second QA system that operates on paragraph-level information retrieval. We compare two configurations of the system using the top-50 and top-80 ranked paragraphs to extract the answer against our threshold-based approach and regression approach that selects the cutoff within the first 250 paragraphs.

uses cosine similarity to score tf-idf-weighted bag-of-word (unigram) vectors. The reader is a modified version of the DrQA document reader with an additional bi-directional attention layer (Seo et al., 2017). We are testing two different configurations[1] of this system: one that selects the top-50 paragraphs and one that selects the top-80 paragraphs against our approach as shown in Tab. 2. We see that, owed to the paragraph-level information retrieval, the number of top-n passages gains even more importance. Both variations of the system outperform a system without adaptive retrieval, which confirms our findings.

6 Conclusion

Our contribution is three-fold. First, we establish that deep question answering is subject to a noise-information trade-off. As a consequence, the number of selected documents in deep QA should not be treated as fixed, rather it must be carefully tailored to the QA task. Second, we propose adaptive schemes that determine the optimal document count. This can considerably bolster the performance of deep QA systems across multiple benchmarks. Third, we further demonstrate how crucial an adaptive document retrieval is in the context of different corpus sizes. Here our adaptive strategy presents a flexible strategy that can successfully adapt to it and, compared to a fixed document count, accomplishes the best performance in terms of regret.

Reproducibility

Code to integrate adaptive document retrieval in custom QA system and future research is freely available at https://github.com/bernhard2202/adaptive-ir-for-qa

Acknowledgments

We thank the anonymous reviewers for their helpful comments. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. Cloud computing resources were provided by a Microsoft Azure for Research award.

[1] Best configurations out of $\{30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$ on SQuAD train split.

References

Petr Baudiš and Jan Šedivý. 2015. Modeling of the question answering task in the yodaqa system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 222–228, Cham. Springer.

Nicholas Belkin. 1993. Interaction with texts: Information retrieval as information-seeking behavior. *Information Retrieval*, 93:55–66.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing*, pages 1533–1544.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the AskMSR question-answering system. In *Empirical Methods in Natural Language Processing*, pages 257–264.

YongGang Cao, Feifan Liu, Pippa Simpson, Lamont Antieau, Andrew Bennett, James J. Cimino, John Ely, and Hong Yu. 2011. Askhermes: An online question answering system for complex clinical questions. *Journal of Biomedical Informatics*, 44(2):277–288.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Annual Meeting of the Association for Computational Linguistics*, pages 1870–1879.

Christopher Clark and Matt Gardner. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.

Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2000. Falcon: Boosting knowledge for answer engines. In *Text REtrieval Conference*, pages 479–488.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems*, 1:1693–1701.

Michael Kaisser and Tilman Becker. 2004. Question answering by searching large corpora with linguistic methods. In *Text REtrieval Conference*.

Bernhard Kratzwald and Stefan Feuerriegel. 2018. Putting question-answering systems into practice: Transfer learning for efficient domain customization. *arXiv preprint arXiv:1804.07097*.

Vanessa Lopez, Victoria Uren, Enrico Motta, and Michele Pasin. 2007. Aqualog: An ontology-driven question answering system for organizational semantic intranets. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):72–105.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Empirical Methods in Natural Language Processing*, pages 1400–1409.

Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems*, 21(2):133–154.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing*, pages 2383–2392.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Dan Shen and Dietrich Klakow. 2006. Exploring correlation of dependency relation paths for answer extraction. In *Annual Meeting of the Association for Computational Linguistics*, pages 889–896.

Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. 2012. Template-based question answering over rdf data. In *Conference on World Wide Web*, page 639.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018. R3: Reinforced ranker-reader for open-domain question answering. In *Conference on Artificial Intelligence*.