A Neural Passage Model for Ad-hoc Document Retrieval

Qingyao Ai(✉), Brendan O'Connor, and W. Bruce Croft

College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA 01003-9264, USA
{aiqy,brenocon,croft}@cs.umass.edu

Abstract. Traditional statistical retrieval models often treat each document as a whole. In many cases, however, a document is relevant to a query only because a small part of it contain the targeted information. In this work, we propose a neural passage model (NPM) that uses passage-level information to improve the performance of ad-hoc retrieval. Instead of using a single window to extract passages, our model automatically learns to weight passages with different granularities in the training process. We show that the passage-based document ranking paradigm from previous studies can be directly derived from our neural framework. Also, our experiments on a TREC collection showed that the NPM can significantly outperform the existing passage-based retrieval models.

Keywords: Passage-based retrieval model · Neural network

1 Introduction

Ad-hoc retrieval refers to a key problem in Information Retrieval (IR) where documents are ranked according to their assumed relevance to the information need of a specific query formulated by users [1]. In the past decades, statistical models have dominated the research on ad-hoc retrieval. They assume that documents are samples of n-grams, and relevance between a document and a query can be inferred from their statistical relationships. To ensure the reliability of statistical estimation, most models treat each document as single piece of text.

There are, however, multiple reasons that motivate us not to treat a document as a whole. First, there are many cases where the document is relevant to a query only because a small part of it contains the information pertaining to the user's need. The ranking score between the query and these documents would be relatively low if we construct the model with statistics based on the whole documents. Second, because reading takes time, sometimes it is more desirable to retrieve a small paragraph that answers the query rather than a relevant document with thousands of words. For instance, we do not need a linux textbook to answer a query "linux copy command".

Given these observations, IR researchers tried to extract and incorporate relevance information from different granularities for ad-hoc document retrieval.

G. Pasi et al. (Eds.): ECIR 2018, LNCS 10772, pp. 537–543, 2018.
https://doi.org/10.1007/978-3-319-76941-7_41

One simple but effective method is to cut long documents into pieces and construct retrieval models based on small passages. Those passage-based retrieval models are able to identify the subtopics of a document and therefore capture the relevance information with finer granularities. Also, they can extract relevant passages from documents and provide important support for query-based summarization and answer sentence generation.

Nonetheless, the development of passage-based retrieval models is limited because of two reasons. First, as far as we know, there is no universal definition of passages in IR. Most previous studies extracted passages from documents with a fixed-length window. This method, however, is not optimal as the best passage size varies according to both corpus properties and query characteristics. Second, aggregating information from passages with different granularities is difficult. The importance of passages depends on multiple factors including the structure of documents and the clarity of queries. For example, Bendersky and Kurland [2] noticed that passage-level information is not as useful on highly homogeneous documents as it is on heterogeneous documents. A simple weighting strategy without considering these factors is likely to fail in practice.

In this paper, we focus on addressing these challenges with a unified neural network framework. Specifically, we develop a convolution neural network that extracts and aggregates relevance information from passages with different sizes. In contrast to previous passage-based retrieval models, our neural passage model takes passages with multiple sizes simultaneously and learns to weight them with a fusion network based on both document and query features. Also, our neural passage model is highly expressive as the state-of-the-art passage-based retrieval models can be incorporated into our model as special cases. We conducted empirical experiments on TREC collections to show the effectiveness of the neural passage model and visualized the network weights to analyze the effect of passages with different granularities.

2 Related Work

Passage Extraction. Previous studies have explored three types of passage definitions: structure-based, topic-based and window-based passages. Structure-based passage extraction identifies passage boundaries with author-provided marking such as empty line, indentation etc. [7]. Topic-based passage extraction, such as TextTiling [3], divides documents into coherent passages with each passage corresponding to a specific topic. Despite its intuitive motivation, this approach is not widely used because identifying topic drift in documents is hard and computationally expensive. Instead, the most widely used methods extract passages with overlapped or non-overlapped windows [10].

Passage-based Retrieval Model. Most passage-based retrieval models in previous studies are unigram models constructed on window-based passages with fixed length. Liu and Croft [5] applied the language modeling approach [6] on overlapped-window passages and ranked documents with their maximum passage language score. Bendersky and Kurland [2] combined passage-level language

models with document-level language models and weighted them according to the homogeneity of each document. To the best of our knowledge, our work is the first study that incorporates a neural network framework for passage-based retrieval models.

3 Neural Passage Model

In this section, we describe how to formulate the passage-based document ranking with a neural framework and aggregate information from passages with different granularities in our neural passage model (NPM). The overall model structure is shown in Fig. 1.

Passage-based Document Ranking. Passage-based retrieval models use passages as representatives for each document and rank documents according to their passage scores. Specifically, given a query q and a passage g extracted with a fixed-length window, the score of g is the maximum log likelihood of observing q given g's unigram language model as

$$\log P(q|g) = \sum_{t \in q} \log P(t|g) = \sum_{t \in q} \log((1-\lambda_c)\frac{tf_{t,g}}{n} + \lambda_c \frac{cf_t}{|C|}) \tag{1}$$

where $tf_{t,g}$ is the count of t in g, cf_t is the corpus frequency of t, $|C|$ is the length of the corpus and λ_c is a smoothing parameter.

Assuming that passages can serve as proxies of documents [2], the ranking score of a document d under the passage-based document ranking framework should be $Score(q,d) = \log \sum_{g \in d} P(g|d) \cdot P(q|g)$. Intuitively, $P(g|d)$ could be a uniform distribution since all passages are extracted following the same methodology. However, averaging passage scores produces poor retrieval performance in practice and the state-of-the-art models adopt a winner-take-all strategy that only uses the best passage to compute document scores [2,5]:

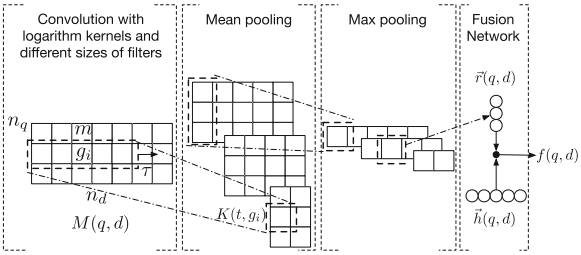

Fig. 1. The structure of the NPM.

$$Score(q,d) = \max_{g \in d} \log P(q|g) \tag{2}$$

Passage Extraction with a Convolution Layer. Given a fixed length window, window-based passages are extracted by sliding the window along the document with fixed step size. Formally, given a document d with length n_d, the set of extracted passages $G(d)$ with window size m and step size τ is

$G(d) = \{g_i | i \in [0, \lfloor \frac{n_d}{\tau} \rfloor]\}$ where g_i represents the ith passage starting from the $i \cdot \tau$th term in d with size m. Let n_q be the length of query q, then the matching of terms in q and d is a matrix $M(q,d) \in \mathbb{R}^{n_q \times n_d}$ in which $M(q,d)_{i,j}$ represents the matching between the ith term in q and the jth term in d. In this work, we define the matching of two terms as a binary variable (1 if the two terms are same and 0 if they are not). Let $K(t, g_i)$ be the score of g_i given term t in q, then the extraction of window-based passages can be achieved with a convolution filter with size m, stride τ and kernel K over $M(q,d)$. Passages with different granularities can be directly extracted with different sizes of filters.

Language Modeling as a Logarithm Kernel. Let $M(t, g_i)$ be the binary matching matrix for term t and g_i with shape $(1, m)$ and $W \in \mathbb{R}^m$, $b_t \in \mathbb{R}$ be the parameters for a logarithm convolution kernel K, then we define the passage model score for q and g_i as

$$Score(q, g_i) = \sum_{t \in q} K(t, g_i) = \sum_{t \in q} \log(W \cdot M(t, g_i) + b_t) \tag{3}$$

Let W be a vector of 1 and b_t be $\frac{\lambda_c \cdot m \cdot cf_t}{(1-\lambda_c)|C|}$, then $K(t, g_i)$ is equal to $\log P(t|g_i)$ in Eq. 1 plus a bias ($\log \frac{m}{1-\lambda_c}$). Thus, the term-based language modeling approach can be viewed as a logarithm activation function over the linear projection of $M(t, g_i)$. Further, if we implement the sum of query term scores with a mean pooling and the winner-take-all strategy in Eq. 2 with a max pooling, then the passage-based document ranking framework can be completely expressed with a three-layer convolution neural network.

Aggregating Passage Models with a Fusion Network. Bendersky and Kurland observed that the usefulness of passage level information varies on different documents [2]. To consider document characteristics, they proposed to combine the passage models with document models using document homogeneity scores $h^{[M]}(d)$ as $Score(q,d) = h^{[M]}(d)P(q|d) + (1 - h^{[M]}(d)) \max_{g \in d} P(q|g)$ where $h^{[M]}(d)$ could be length-based ($h^{[length]}$), entropy-based ($h^{[ent]}$), inter-passage ($h^{[intPsg]}$) or doc-passage ($h^{[docPsg]}$):

$$\begin{aligned} h^{[length]}(d) &= 1 - \frac{\log n_d - \min_{d_i \in C} \log n_{d_i}}{\max_{d_i \in C} \log n_{d_i} - \min_{d_i \in C} \log n_{d_i}} \\ h^{[ent]}(d) &= 1 + \frac{\sum_{t' \in d} P(t'|d) \log(P(t'|d))}{\log n_d} \\ h^{[intPsg]} = \frac{2}{\lceil \frac{n_d}{\tau} \rceil (\lceil \frac{n_d}{\tau} \rceil - 1)} \sum_{i<j; g_i, g_j \in d} \cos(g_i, g_j), &\quad h^{[docPsg]} = \frac{1}{\lceil \frac{n_d}{\tau} \rceil} \sum_{g_i \in d} \cos(d, g_i) \end{aligned} \tag{4}$$

where $\cos(d, g_i)$ is the cosine similarity between the tf.idf vector of d and g_i.

Inspired by the design of homogeneity scores and studies on query performance prediction [9], we propose a fusion network that aggregates scores from passages according to both document properties and query characteristics. We extract features for queries and concatenate them with the homogeneity features to form a fusion feature vector $\boldsymbol{h}(q,d)$. For each query term, we

extract their inverse document/corpus frequency and a clarity score [9] defined as $SCQ_t = (1+\log(cf_t))\log(1+idf_t)$ where idf_t is the inverse document frequency of t. For each feature, we compute the sum, standard deviation, maximum, minimum, arithmetic/geometric/harmonic mean and coefficient of variation for $t \in q$. We also include a list feature as the average scores of top 2,000 documents retrieved by the language modeling approach. Suppose that $\boldsymbol{h}(q,d) \in \mathbb{R}^{\beta}$ and let $\boldsymbol{r}(q,d) \in \mathbb{R}^{\alpha}$ be a vector where each dimension denotes a score from one convolution filter, then the final ranking score $f(q,d)$ is computed as

$$Score(q,d) = f(q,d) = \tanh\left(\boldsymbol{r}(q,d)^T \cdot \phi(\boldsymbol{h}(q,d)) + b_R\right) \tag{5}$$

where $\phi(\boldsymbol{h}(q,d)) = \frac{\exp(\boldsymbol{W}_R^i \cdot \boldsymbol{h}(q,d))}{\sum_{j=1}^{\alpha} \exp(\boldsymbol{W}_R^j \cdot \boldsymbol{h}(q,d))}$ and $W_R \in \mathbb{R}^{\alpha \times \beta}$, $b_R \in \mathbb{R}$ are parameters learned in the training process.

Table 1. The performance of passage-based retrieval models. $*$, $+$ means significant differences over MSP[base] and MSP[docPsg] with passage size (150, ∞) respectively.

	MAP	NDCG@20	Precison@20	MAP	NDCG@20	Precison@20
	Passage Size (50, ∞)			Passage Size (150, ∞)		
MSP[base]	0.193	0.317	0.288	0.207	0.335	0.302
MSP[length]	0.210	0.333	0.298	0.223*	0.355*	0.315*
MSP[ent]	0.209	0.338	0.304	0.216*	0.349*	0.314*
MSP[interPsg]	0.204	0.329	0.296	0.215*	0.346*	0.310*
MSP[docPsg]	0.206	0.331	0.296	0.226*	0.362*	0.312*
	Passage Size (50, 150, ∞)					
NPM[doc]	0.255^{*+}	0.412^{*+}	0.366^{*+}	-	-	-
NPM[query]	**0.256**$^{*+}$	**0.416**$^{*+}$	**0.369**$^{*+}$	-	-	-
NPM[doc+query]	0.255^{*+}	0.413^{*+}	0.367^{*+}	-	-	-

4 Experiment and Results

In this section, we describe our experiments on Robust04 with 5-fold cross validation [4]. For efficient evaluation, we conducted an initial retrieval with the query likelihood model [6] and performed re-ranking on the top 2,000 documents. We reported MAP, NDCG@20, Precision@20 and used Fisher randomization test [8] ($p < 0.05$) to measure the statistical significance. Our baselines include the max-scoring language passage model [5](MSP[base]) and the state-of-the-art passage-based retrieval model with passage weighting [2] – the MSP with length scores (MSP[length]), the MSP with entropy scores (MSP[ent]), the MSP with inter-passage scores (MSP[interPsg]) and the MSP with doc-passage scores (MSP[docPsg]). We follow the same parameter settings used by Bendersky and Kurland [2] and tested all models with passage size 50 and 150 separately. We used filters with length 50, 150 and ∞ for NPMs and set τ as the half of the filter

lengths. The filter with length 50 extracts the same passages used in MSP models with passage size 50, and the filter with length ∞ treats the whole document as a single passage. Notice that the MSP with passage weighting [2] also uses sizes 50 (or 150) and ∞ to combines the scores of passages and the whole document. We tested the NPMs with document homogeneity features (NPM[doc]), query features (NPM[query]) and both (NPM[doc+query]). Due to the limit of Robust04, we only have 249 labeled queries, which are far from enough to train a robust convolution kernel with hundreds of parameters. Therefore, we fixed the convolution kernels as discussed in Sect. 3.

Overall Performance. Table 1 shows the results of our baselines and the NPM models with passage size 50 and 150. As we can see, the variations of MSP significantly outperformed MSP[base] with the same passage size, and the MSP models with passage size 150 performed better than MSP with passage size 50. Compared to MSP models, the NPM models showed superior performance on all reported metrics. As discussed in Sect. 3, MSP models can be viewed as special cases of the NPM with predefined parameters. With passage size 50, the MSP[base] model is actually a NPM model with filter length 50 and no fusion layer; and the MSP with homogeneity weighting is a NPM model with filter lengths 50, ∞ and a linear fusion with document homogeneity scores. From this perspective, the NPM model is more powerful than MSP models as it automatically learns to weight passages according to document/query features.

Weights of Passages. Figure 2 shows the means of passage weights $\phi(\boldsymbol{h}(q,d))$ on all query-doc pairs for NPM[query], NPM[doc] and NPM[doc+query]. In our experiments, the passages with size ∞ are the most important passages in the NPMs, but the scores from smaller passages also impact the final ranking. Although the MAP of the NPM[query] and NPM[doc] are close, their passage weights are different and they performed differently on 211 of 249 queries on Robust04. This indicates that, when evaluating a document with respect to multiple queries, all passages could be useful to determine the document's relevance; when evaluating multiple documents with respect to one query, models with ∞ passage size are more reliable in discriminating relevant documents from irrelevant ones.

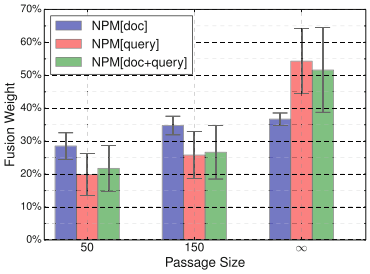

Fig. 2. The fusion weights for passages in the NPM averaged over query-doc pairs.

5 Conclusion and Future Work

In this paper, we proposed a neural network model for passage-based ad-hoc retrieval. We view the extraction of passages as a convolution process and develop a fusion network to aggregate information from passages with different granularities. Our model is highly expressive as previous passage-based retrieval models can be incorporated into it as special cases. Due to the limit of our data, we used binary matching matrix and deprived the freedom of the NPM to learn convolution kernels automatically. We will explore its potential to discover new matching patterns from more complex signals and heterogeneous datasets.

Acknowledgments. This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1160894. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

1. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern Information Retrieval, vol. 463. ACM Press, New York (1999)
2. Bendersky, M., Kurland, O.: Utilizing passage-based language models for document retrieval. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 162–174. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78646-7_17
3. Hearst, M.A.: Texttiling: a quantitative approach to discourse segmentation. Technical report, Citeseer (1993)
4. Huston, S., Croft, W.B.: A comparison of retrieval models using term dependencies. In: CIKM 2014, pp. 111–120. ACM (2014)
5. Liu, X., Croft, W.B.: Passage retrieval based on language models. In: CIKM 2002, pp. 375–382. ACM (2002)
6. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: SIGIR 1998, pp. 275–281. ACM (1998)
7. Salton, G., Allan, J., Buckley, C.: Approaches to passage retrieval in full text information systems. In: SIGIR 1993, pp. 49–58. ACM (1993)
8. Smucker, M.D., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: CIKM 2007, pp. 623–632. ACM (2007)
9. Zhao, Y., Scholer, F., Tsegay, Y.: Effective pre-retrieval query performance prediction using similarity and variability evidence. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 52–64. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78646-7_8
10. Zobel, J., Moffat, A., Wilkinson, R., Sacks-Davis, R.: Efficient retrieval of partial documents. Inf. Process. Manag. **31**(3), 361–377 (1995)