A Comparison Between Term-Based and Embedding-Based Methods for Initial Retrieval

Tonglei Guo(✉), Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng

CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
{guotonglei,fanyixing}@software.ict.ac.cn,
{guojiafeng,lanyanyan,junxu,cxq}@ict.ac.cn

Abstract. The initial retrieval stage of information retrieval aims to generate as many relevant candidate documents as possible in a simple yet efficient way. Traditional term based retrieval methods like BM25 deal with the problem based on Bag-of-Words (BoW) representation, thus they only focus on exact matching (i.e., syntactic) and lack the consideration for semantically related words. That causes the typical vocabulary mismatch problem and the reduction of performance in terms of recall. The advance of distributed representation (i.e., embedding) of words and documents provides an efficient way to measure the semantic relevance between words. Since embedding can alleviate the vocabulary mismatch problem, it is suitable for the initial retrieval task. We conduct several experiments to compare term based models with embedding based models in terms of recall. We compare above two branches of the initial retrieval models on three representative retrieval tasks (Web-QA, Ad-hoc retrieval and CQA respectively). The results show that embedding based method and term based method are complementary for each other and higher recall can be achieved by combining the above two types of models based on scores or ranking position. We find that combination of the two types of the models based on ranking position usually perform better than combination based on score. Furthermore, since queries and documents are in different forms for diverse application scenarios, it can be observed that the relative performance of the two types are almost same but the absolute performance are significant different regarding to distinct scenarios.

Keywords: Initial retrieval · Embedding representation · Recall

1 Introduction

The process of information retrieval can typically divide into two stages, namely the initial retrieval stage and the re-ranking stage [5]. The initial retrieval stage aims to retrieve a small candidate set containing as many relevant documents as possible. It is required to achieve high recall in a simple yet efficient way. Then,

S. Zhang et al. (Eds.): CCIR 2018, LNCS 11168, pp. 28–40, 2018.
https://doi.org/10.1007/978-3-030-01012-6_3

the re-ranking stage applies complicated methods like various deep models to rank the candidate document according to their relevance with the given query. In this paper, we focus on the initial retrieval stage.

Traditional initial retrieval models use term based methods for high efficiency. Term based models are relied on Bag-of-Words (BoW) representation and assumes words in documents are independent from others, the measurement of relevance relies on exact matching (i.e., syntactic or terms counting) of words. But relevant texts sometimes can be expressed in totally different words, e.g., "How bad is the new book by J.K. Rowling?" and "How is the new Harry Potter book Harry Potter and the Cursed Child?" are related obviously but they share little common words. This causes the typical problem in IR called *vocabulary mismatch* and leads to many relevant documents failed to be retrieved in this stage.

The advance of distributed models [14,17,20] provides an efficient way to represent queries and documents semantically. These models can represent a single word or a piece of text by a vector in a continuous semantic vector space, referred to as "embedding". These embedding representations have shown promising results in NLP, which inspires the application in the initial retrieval stage to capture the semantic relevance.

In this paper, we explore term based models and embedding based models in the initial retrieval step. We choose BM25 and language model to represent for term based models. Existing embedding based models can be divided into aggregated distributed representation and paragraph vector representation. We use BoWE model and an enhanced model based on PV to represent two parts. Furthermore, we explore the hybrid models which combine two branches of models to take both exact matching and semantic relevance into account. We evaluate the recall of each model in three typical IR applications: Ad-hoc retrieval, Web-QA and community-based question answering (CQA). We give brief introduction of these applications and describe the characteristic of different scenarios. We design experiments on three benchmark datasets respectively representing three scenarios and do some analysis according to the results. The performance is measured in terms of recall concerned in the initial retrieval step.

In brief, here are some take-away conclusions based on our experimental results:

1. Considering the performance in terms of recall for each method independently, embedding based methods can't outperform term based methods, but it can be seen that cases where they perform well are not overlapped, in other words, they are complementary to each other.
2. Comparing the two embedding based methods, the BoWE has higher recall than the method of PV. There is little difference between the recall of two embedding based methods in CQA task. But when it comes to Ad-hoc retrieval and Web-QA, PV shows poor performance than BoWE.
3. Usually the combination of the term based methods with embedding based methods can outperform term based methods. Moreover, the hybrid of two branches of models based on ranking position usually do better than the combination based on score.

The rest of this paper is organized as follows: In Sect. 2, we discuss some related work regrading to the initial retrieval step. In Sect. 3, we describe the two branches of the initial retrieval model in details. Then we show our experimental results and do some analysis in Sect. 4.

2 Related Work

In this section, we overview the researches related to our works in three areas: term based models, embedding based models and combination models for the initial retrieval step.

2.1 Information Retrieval Process

Information retrieval can be typically divided into two stages [5], namely the initial retrieval stage and the re-ranking stage. Since modern IR system usually need to deal with a huge amount of data, it is impossible to rank all documents given a query. The initial retrieval stage aims to select a small subset of documents containing as many relevant documents as possible (i.e., high recall) in an efficient way. Traditional initial retrieval models are term based models like BM25 [23] since their simplicity and efficiency. Modern IR system achieved this step by building a symbolic based inverted index and a search scheme. On the other hand, the re-ranking stage focus on ranking the above candidate documents according to the relevance of the given query as precise as possible (i.e., high precision). Since the size of the subset is relative small and the requirement of precision, so more complicated models are used in re-ranking stage, for example, various learning to rank algorithms [2,3] and deep learning models [8,18].

2.2 Embedding

The revival and new development of neural networks in natural language processing (NLP) has brought the attention of using these techniques in information retrieval tasks. Existing works of applying neural networks in IR mainly consists of two branches, the first one is utilizing embeddings to represent query and document [9,25], the other one is using deep models to learn representations of query and document [11]. Most deep learning models are complicated so usually they are explored in the re-ranking stage. While the embeddings can be pre-trained and the relevance calculation is simplified to the vector similarity, so it is more promising in the initial retrieval stage.

The embeddings of words and documents in a low-dimensional vector space have attracted many researchers. The ability of embeddings which can encode semantic and syntactic relations between words and documents seems benefits retrieval models in semantic matching. Existing works generally utilize word2vec [17] to obtain pre-trained word embeddings and then utilize them in retrieval tasks [9,25]. Besides above methods which combines pre-trained word embeddings, there are some methods learn embeddings from scratch directly.

Le et al. [14] propose doc2vec which learns words and documents embeddings representations together. The learned representation can be used to measure the relevance between queries and documents by the similarity of embedding vectors.

3 Initial Retrieval Models

Term based models calculate the relevance based on the occur frequencies of query words, i.e., exact matching which causes mismatch of many semantic related word pairs, e.g., "metro" and "subway". That leads to the typical vocabulary mismatch problem in IR and degrades the performance of retrieval models. The recently proposed word embedding, which can capture semantic relatedness between words, have attracted a lot of attentions in many NLP tasks. The relevance can be calculated efficiently from vector similarity so it is worthwhile to explore the embedding based initial retrieval models.

In this section, we explore the embedding based models in the initial retrieval step and use them to enhance the term based models. We start with the term based models in Sect. 3.1 and then introduce two kinds of embeddings based models in detail. In Sect. 3.3, we describe the hybrid models of above models in two ways. At last, we give brief introduction of three typical IR applications where above models are used in.

3.1 Term Based Initial Retrieval Models

Most term based models for the initial retrieval step use BoW representation based on the assumption that words are independent from others. In this way, each word can be viewed as a dimension in a orthogonal semantic vector space where each query or document is represented as a point. A query and a document which is close to each other will be considered as relevant pairs.

BM25. BM25 [23] is a representative state-of-the-art retrieval model which based on the probabilistic ranking principle. Both query and document are represented as a vector using bag-of-words assumption. BM25 calculate the relevance score of the document based on the query term frequencies and document length: $BM25(q,d) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f_i \cdot (k_1+1)}{f_i + k_1 \cdot (1-b+b \cdot \frac{dl}{avgdl})}$. Where k and b are tunable hyper-parameters.

Language Model. Language model [22,26,27] is based on a probabilistic model which predict the probability that a query q is generated by a document d. According to the Bayes'formula, $p(d|q) \propto p(q|d) \cdot p(d)$. Thus the relevance of the given query q and a candidate document d is measured by the conditional probability $p(d|q)$. Language models usually have various smoothing methods [22,26,27] to overcome the zero probability of unseen words and data sparseness.

3.2 Embedding Based Initial Retrieval Models

Distributed representation, also named embedding, which is learned based on the distributional hypothesis [6,10], has achieved a lot of success in natural language processing. It has been found that the distributed representation of documents can well capture the semantic relatedness of documents [17]. Thus, it is straightforward to employ document embedding to build the retrieval model in the initial retrieval stage. Embedding based retrieval models can be categorized into two classes according to the way of constructing document embedding, namely aggregated distributional representation and paragraph vector representation. In the follows, we will describe each model in detail.

Aggregated Distributed Representation. Aggregated distributed representation views a word in a text as the basic unit and aggregated embeddings of these words in a specific way to obtain the final document representation. There have been a large number of works [9,25] which studied how to integrate the word embeddings to obtain the document representation in an effective way. In this paper, we directly utilized the simple but effective aggregated model which is based on weighted sum [9]. Denote D is the collection of all documents and d is a document in D. BoWE representation views a documents as a bag of words in d. Given the word embedding matrix $W \in R^{K \times |V|}$ for a finite size V of vocabulary consists of all words from the corpus. Each column $W^i \in R^k$ is the K dimensional word embedding of the i-th words. Then the document d can be expressed as $d = w_1^d, w_2^d, \ldots, w_n^d$, n is the number of words in d and w_i^d denotes the embedding of i-th words. We represent the document d as a fixed length vector in following way:

$$\overrightarrow{d} = IDF_1 \cdot \overrightarrow{w_1} + IDF_2 \cdot \overrightarrow{w_2} + \cdots + IDF_n \cdot \overrightarrow{w_n} \tag{1}$$

Paragraph Vector Representation. To utilize the information of words order, there comes another embeddings way which views a piece of paragraph as the basic unit [14]. In this paper we use an enhanced model based on the PV-DBOW model [14] which predict unigram, bigram and trigram words randomly sampled from the paragraph. Each n-gram of words is represented by a distinct embeddings.

3.3 Hybrid Models

Term based models lost semantic matching of words which limits the performance of models. To overcome the defect of vocabulary mismatch problem and consider semantic relevance, we propose two hybrid models to combine term based models with embedding based models. Let $f_{BoW}(q, d)$ denote a term based model and $f_{emb}(q, d)$ denote an embedding based model for measuring the relevance of a given query q and a candidate document d.

Score-Based Hybrid Model. This model utilize the score of two branches of models directly. It calculates the final score while linearly combining the score results of two branches models.

$$f_s(q,d)_{score} = (1-\lambda) \cdot f_{BoW}(q,d)_{score} + \lambda \cdot f_{emb}(q,d)_{score} \quad (2)$$

Rank-Based Hybrid Model. This model is based on the result of ranking positions obtained by the two branches of models. It ignores the exact score value but focus on the positions in the ranking lists. The scores used for combination is the reciprocal of the position in the resulting ranking lists: $s(q,d) = \frac{1}{r}$.

$$f_s(q,d)_{rank} = (1-\lambda) \cdot f_{BoW}(q,d)_{rank} + \lambda \cdot f_{emb}(q,d)_{rank} \quad (3)$$

3.4 Typical IR Applications

Information retrieval (IR) has various applications, such as Ad-hoc retrieval, Web-Question-Answering (Web-QA) and community question answering (CQA) etc. This paper focus on three typical IR applications, so in this section, we give brief review of these three IR applications.

Ad-hoc Retrieval. The core problem of Ad-hoc retrieval [11,16] is to measure the relevance for a document given the question. The task of ad-hoc retrieval is to fetch the relevant candidate documents and rerank them according to the relevance given a query. Usually the documents in the dataset are stable while the queries are various with different users. The initial retrieval step in ad-hoc retrieval is required to return as many related documents as possible and leave the re-ranking to the next step.

Web-QA. Web-QA [7,15,21] is based on IR models which relies on the massive data formalized as texts on the web. The task of web-QA is to use information retrieval techniques to extract the relevant documents on web given a specific question. A web-QA system would firstly process the question in natural language to understand the question properly like the answer type etc. and then, formulates a query for a search engine to retrieve relevant documents. Finally the correct answer would be extracted from the most related documents. For the initial retrieval step, the web-QA models only focus on retrieve relevant documents which might contain correct answer according to the question.

CQA. Community question answering (CQA) [1,4,19] is increased in popularity with the advance of many community forums. Its task is to find the correct answer of the given question from the dataset contributed by all the users. Different from the web-QA system which tries to find answer directly, a CQA system would check if there is a duplicated question and return the answer of this question. If there isn't any similar question with the given one, then the system leave it to other users to answer the question and allow them rate the answers. In this paper, we focus on the initial retrieval step which tries to find similar questions with the given question.

4 Experiement

In this section, we conduct experiments in three typical IR application: Ad-hoc retrieval, Web-QA and community-based question answering(CQA). We choose

Table 1. The feature of three typical IR applications.

Application	Text1	Text2	Type
Ad-hoc retrieval	"International organized crime"	"Guatemalan President has sent a letter to U.S. Secretary of State Warren Christopher to promote the visit of U.S. tourists to Guatemala, saying the recent acts of violence against two U.S. citizens were isolated incidents. Due to those acts of violence, the U.S. Government has issued a travel warning advising travelers not to visit Guatemala or to postpone their trips to that country, except for emergency purposes."	Short-long
Web-QA	"How to remove tree sap from car"	"I have had good luck with commercial products available at auto supply stores for just a few dollars below are some additional suggestions if you want to try DIY"	Medium-medium
CQA	"Why do rockets looks white?"	"Why are rockets and boosters painted white?"	Short-short

three benchmark collections for each application scenario and compare performance of models introduced in Sect. 3 respectively.

4.1 Datasets

For the Ad-hoc retrieval application, we choose a subset of Robust04. Robust04 is a dataset related to news and its topics are chosen from TREC Robust Track 2004. We use the description of each topic and a subset of these documents which has a headline. As for Web-QA application, we use Yahoo-QA dataset. The data is a subset of the Yahoo! Answers corpus from a 10/25/2007 dump. It is a small subset of the questions selected for their linguistic properties (for example they all start with "how $\{to|do|did|does|can|would|could|should\}$". For CQA application, we conduct our experiment on Quora dataset. Quora dataset consists of over 400,000 lines of potential question duplicate pairs. For all the datasets in our experiments, both queries and documents are stemmed using the Krovetz stemmer [13]. Stopwords is removed according to the INQUERY stop list. The statistics of these datasets are listed in Table 1.

Table 2. Statistics of the datasets used in this study.

	Yahoo-QA	Robust04	Quora
Vocabulary	53K	0.5M	77K
Document count	0.5M	0.34M	0.54M
Collection length	4.5M	474M	21M
Query count	9401	233	0.54M

4.2 Experimental Settings

Baseline Methods. We adopt BM25 as our baselines. We use the Indri to build index for documents and adopt the BM25TF achieved in the Indri as our baselines. We set the parameters as default in the Indri complemention, i.e. $k_1 = 1.2$ and $b = 0.75$ except for Robust04 datasets. Following the study of [12], we set $k_1 = 0.708$ and $b = 0.325$ for Robust04 dataset.

LM_DIR. The language model implemented in the Indri is based on a combination of the language modeling [22] and inference network [24] retrieval frameworks. Here we choose Dirichlet as smoothing method. We use default parameters, i.e., $\mu = 2500$ for Dirichlet.

Word Embeddings. To avoid randomness caused by out-of-vocabulary words, we adopt corpus-specific word embeddings in our experiment. We train word vectors using CBOW model [17] with negative sampling. We use Word2Vec[1] training 300-dimension word embeddings. We set the context window size to 10 and use 10 negative samples and the sampling threshold is 10^{-4} as default. The minimum count is set to 1 so that there won't be any word out-of-vocabulary.

Paragraph Vectors. The dimension of PV is 300. We use 6 negative samples and the sampling threshold is 10^{-4} as default and minimum count is set to 1.

Hybrid Models. We choose BM25 represents term based models, so there are totally four hybrid models namely $BM25 + BoWE_{score}$, $BM25 + BoWE_{rank}$, $BM25 + PV_{score}$ and $BM25 + PV_{rank}$ respectively. For coefficient used in each hybrid model, we choose the best from the range $(0, 1)$ increased by 0.05 each time according to experiments. Different hybrid model prefer different coefficient but the tendency are same. We show the coefficient tendency of the $BM25 + BoWE_{score}$ in Fig. 1.

Evaluation Measures. Since the initial retrieval step focus on retrieving as many related candidate documents as possible, thus we measure the model's performance in terms of recall@K only. We set different value of K for each datasets. Specifically, we set K to 500, 1000 and 100 for Yahoo-QA, Robust04 and Quora respectively.

[1] http://nlp.stanford.edu/projects/glove/.

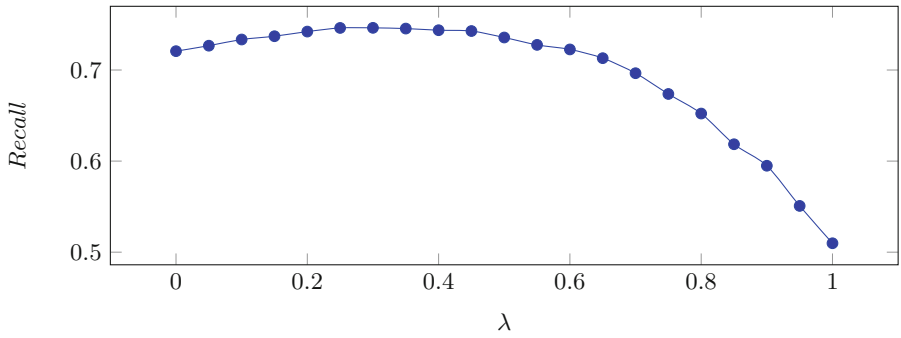

Fig. 1. Coefficient tendency of the $BM25 + BoWE_{score}$ Model

4.3 Retrieval Performance and Analysis

In this section we present our experimental results over three datasets. Note that we choose BM25TF as our baseline and "win", "tie", "lose" in each table mean that the number of cases win/tie/lose compared to baseline model.

Table 2 shows the result of models on Robust04 dataset which is corresponding to the application of Ad-hoc retrieval. In Robust04 dataset the queries is in short description form while the documents are long texts. The result shows that except for PV, all these models perform better than QA task. Since the relevant documents are required to describe the given topic, so the whole passage is taken into consideration. Although traditional models calculate relevance for the whole documents but due to the drawback of the BoW method, they suffer the vocabulary mismatch problem and loss semantic matching information, so the performance is not so well. Embedding based methods still can't do well in long texts but it do help in improving semantic relevance.

Table 3. Comparison of different initial retrieval methods over Robust04 for retrieving 1000 documents.

Robust04 collection (Recall@1000)					
Retrieval type	Retrieval method	Recall (%)	Win	Tie	Lose
Term based	BM25	72.07	–	–	–
	LM_DIR	72.52	54	126	53
Embedding based	BoWE	50.98	34	30	169
	PV	1.67	1	4	228
Hybrid	$BM25 + BoWE_{score}$	74.63	93	112	28
	$BM25 + PV_{score}$	72.67	55	166	12
	$BM25 + BoWE_{rank}$	74.41	78	129	26
	$BM25 + PV_{rank}$	71.69	5	186	42

Table 3 shows the result of different models on Yahoo-QA dataset which is corresponding to the application of QA. The problem of vocabulary mismatch and the requirement of finding the semantic relation between the queries and documents become more obviously in the QA scenario. Both queries and documents may be long texts in the QA tasks as in Yahoo-QA dataset. So the relevance measurement becomes more complicated. As the experimental result shows, the recall of each model stay in relative low level. Despite that, the performance of two models only based on embedding lag far behind traditional models, especially for PV. The reason may be that embedding based method aggregate whole text information together in a rough way which may lose many useful information for matching. Besides, since the length of queries is long, it calls for a method to understand the intension of the queries which is important in finding answers, but all these models fails in that. Further more, the answer for a given question is just a piece of words in document in most cases, so it is not proper to consider the relevance of whole documents. More Sophisticated models need to be used in this task.

Table 4 shows the result of different models on Quora dataset which is corresponding to the application of CQA. CQA tries to find the relevant question pairs, so the length of texts is relatively short. What's more, for Quora dataset, the problem of vocabulary mismatch in not so obviously, so even simply by traditional models based on BoW can gain pretty well performance. It can also be seen that methods based on embedding perform well too. When combined with traditional models, they contribute more (Table 5).

Table 4. Comparison of different initial retrieval methods over Yahoo-QA for retrieving 500 documents.

Yahoo-QA collection (Recall@500)					
Retrieval type	Retrieval method	Recall (%)	Win	Tie	Lose
Term based	BM25TF	55.45	–	–	–
	LM_DIR	53.98	156	8661	583
Embedding based	BoWE	33.25	859	3804	4737
	PV	8.84	264	1968	7168
Hybrid	$BM25 + BoWE_{score}$	56.61	825	8073	502
	$BM25 + PV_{score}$	55.02	231	8843	326
	$BM25 + BoWE_{rank}$	56.82	900	7953	547
	$BM25 + PV_{rank}$	55.52	345	8745	310

Table 5. Comparison of different initial retrieval methods over Quora for retrieving 100 documents.

Quora collection (Recall@100)					
Retrieval type	Retrieval method	Recall (%)	Win	Tie	Lose
Term based	BM25TF	94.27	–	–	–
	LM_DIR	87.66	596	134290	14708
Embedding based	BoWE	89.69	6279	128535	14780
	PV	84.96	4615	121112	23867
Hybrid	$BM25 + BoWE_{score}$	95.42	5112	143408	1074
	$BM25 + PV_{score}$	94.66	4422	142976	2196
	$BM25 + BoWE_{rank}$	95.49	5453	142351	1790
	$BM25 + PV_{rank}$	94.97	4153	143548	1893

Now let's overview the all experimental results to draw some general conclusions.

First of all, comparing the performance of various traditional methods, we can see that in the initial retrieval step, different traditional methods perform nearly equally. But BM25 outperform language model with different smoothing methods slightly in general. The performance of these models are equal on most cases.

Next, when it comes to the methods based on embeddings, we can see that both two methods can't outperform the traditional methods. PV perform extremely poor in long texts scenario. Similarly, BoWE also perform badly in long texts scenario. Long texts contain too many information to be considered so it is hard to aggregate all these information in a simple way like PV or BoWE. Although embedding based method perform pretty well in many NLP tasks, in IR tasks there need some enhancement to fit the special requirements.

Finally, we can find that embedding based method do better in semantic matching. The retrieval results of embedding based methods are quite different from traditional methods. In other words, their advantages are not overlapped. So when combine the traditional models with embedding based models simply through linear combination, the performance still can be improved. Through the combination, we add some semantic relevance information into merely words co-occurrence (i.e., exact matching). What's more, the combination by ranking position is better than combination through score in most cases.

5 Conclusion

In this paper, we conduct experiments on different IR application tasks and draw the conclusion that embedding based models do help in improving the recall of traditional initial retrieval models. We confirm that embedding based methods BoWE representation and Paragraph Vector can help traditional model in the

cases that queries and documents are related in semantic level which share little common words. Through two combination methods, we illustrate that traditional retrieval models combined with embedding based models can perform better than the single one.

For future work, we would like to find a better method to capture semantic relevance information and further more, come up with more effective method to combine the advantage of exact matching models and semantic matching models.

References

1. Blooma, M.J., Kurian, J.C.: Research issues in community based question answering. In: PACIS, p. 29 (2011)
2. Burges, C., et al.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 89–96. ACM (2005)
3. Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th International Conference on Machine Learning, pp. 129–136. ACM (2007)
4. Carmel, D., Mejer, A., Pinter, Y., Szpektor, I.: Improving term weighting for community question answering search using syntactic analysis. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, pp. 351–360. ACM (2014)
5. Dang, V., Bendersky, M., Croft, W.B.: Two-stage learning to rank for information retrieval. In: Serdyukov, P. (ed.) ECIR 2013. LNCS, vol. 7814, pp. 423–434. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36973-5_36
6. Firth, J.R.: A synopsis of linguistic theory, 1930–1955. In: Studies in Linguistic Analysis (1957)
7. Galitsky, B.: Natural language question answering system: technique of semantic headers. Advanced Knowledge International (2003)
8. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 55–64. ACM (2016)
9. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: Semantic matching by non-linear word transportation for information retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 701–710. ACM (2016)
10. Harris, Z.S.: Distributional structure. Word **10**(2–3), 146–162 (1954)
11. Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 2333–2338. ACM (2013)
12. Huston, S., Croft, W.B.: Parameters learned in the comparison of retrieval models using term dependencies. Ir, University of Massachusetts (2014)
13. Krovetz, R.: Viewing morphology as an inference process. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and development in Information Retrieval, pp. 191–202. ACM (1993)
14. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)

15. Lin, J.: The web as a resource for question answering: perspectives and challenges. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002). Citeseer (2002)
16. Lu, Z., Li, H.: A deep architecture for matching short texts. In: Advances in Neural Information Processing Systems, pp. 1367–1375 (2013)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
18. Pang, L., Lan, Y., Guo, J., Xu, J., Xu, J., Cheng, X.: DeepRank: a new deep architecture for relevance ranking in information retrieval. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 257–266. ACM (2017)
19. Patra, B.: A survey of community question answering. arXiv preprint arXiv:1705.04009 (2017)
20. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
21. Perera, R.: IPedagogy: question answering system based on web information clustering. In: 2012 IEEE Fourth International Conference on Technology for Education (T4E), pp. 245–246. IEEE (2012)
22. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 275–281. ACM (1998)
23. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Croft, B.W., van Rijsbergen, C.J. (eds.) SIGIR '94, pp. 232–241. Springer, New York (1994). https://doi.org/10.1007/978-1-4471-2099-5_24
24. Turtle, H., Croft, W.B.: Evaluation of an inference network-based retrieval model. ACM Trans. Inf. Syst. (TOIS) **9**(3), 187–222 (1991)
25. Vulić, I., Moens, M.-F.: Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 363–372. ACM (2015)
26. Zhai, C., Lafferty, J.: Two-stage language models for information retrieval. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 49–56. ACM (2002)
27. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: ACM SIGIR Forum, vol. 51, pp. 268–276. ACM (2017)