A Deep Look into Neural Ranking Models for Information Retrieval

Jiafeng Guo[a,b], Yixing Fan[a,b], Liang Pang[a,b], Liu Yang[c], Qingyao Ai[c], Hamed Zamani[c], Chen Wu[a,b], W. Bruce Croft[c], Xueqi Cheng[a,b]

[a] *University of Chinese Academy of Sciences, Beijing, China*
[b] *CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*
[c] *Center for Intelligent Information Retrieval, University of Massachusetts Amherst, Amherst, MA, USA*

Abstract

Ranking models lie at the heart of research on information retrieval (IR). During the past decades, different techniques have been proposed for constructing ranking models, from traditional heuristic methods, probabilistic methods, to modern machine learning methods. Recently, with the advance of deep learning technology, we have witnessed a growing body of work in applying shallow or deep neural networks to the ranking problem in IR, referred to as neural ranking models in this paper. The power of neural ranking models lies in the ability to learn from the raw text inputs for the ranking problem to avoid many limitations of hand-crafted features. Neural networks have sufficient capacity to model complicated tasks, which is needed to handle the complexity of relevance estimation in ranking. Since there have been a large variety of neural ranking models proposed, we believe it is the right time to summarize the current status, learn from existing methodologies, and gain some insights for future development. In contrast to existing reviews, in this survey, we will take a deep look into the neural ranking models from different dimensions to analyze their underlying assumptions, major design principles, and learning strategies. We compare these models through benchmark tasks to obtain a comprehensive empirical understanding of the existing techniques. We will also discuss what is missing in the current literature and what are the promising and desired future directions.

Preprint submitted to Journal of Information Processing and ManagementNovember 17, 2020

Keywords: neural ranking model, information retrieval, survey
2010 MSC: 00-01, 99-00

1. Introduction

Information retrieval is a core task in many real-world applications, such as digital libraries, expert finding, Web search, and so on. Essentially, IR is the activity of obtaining some information resources relevant to an information need from within large collections. As there might be a variety of relevant resources, the returned results are typically ranked with respect to some relevance notion. This ranking of results is a key difference of IR from other problems. Therefore, research on ranking models has always been at the heart of IR.

Many different ranking models have been proposed over the past decades, including vector space models [1], probabilistic models [2], and learning to rank (LTR) models [3, 4]. Existing techniques, especially the LTR models, have already achieved great success in many IR applications, e.g., modern Web search engines like Google[1] or Bing[2]. There is still, however, much room for improvement in the effectiveness of these techniques for more complex retrieval tasks.

In recent years, deep neural networks have led to exciting breakthroughs in speech recognition [5], computer vision [6, 7], and natural language processing (NLP) [8, 9]. These models have been shown to be effective at learning abstract representations from the raw input, and have sufficient model capacity to tackle difficult learning problems. Both of these are desirable properties for ranking models in IR. On one hand, most existing LTR models rely on hand-crafted features, which are usually time-consuming to design and often over-specific in definition. It would be of great value if ranking models could learn the useful ranking features automatically. On the other hand, relevance, as a key notion in IR, is often vague in definition and difficult to estimate since relevance judgments are based on a complicated human cognitive process. Neural models

[1] http://google.com
[2] http://bing.com

with sufficient model capacity have more potential for learning such complicated tasks than traditional shallow models. Due to these potential benefits and along with the expectation that similar successes with deep learning could be achieved in IR [10], we have witnessed substantial growth of work in applying neural networks for constructing ranking models in both academia and industry in recent years. Note that in this survey, we focus on neural ranking models for textual retrieval, which is central to IR, but not the only mode that neural models can be used for [11, 12].

Perhaps the first successful model of this type is the Deep Structured Semantic Model (DSSM) [13] introduced in 2013, which is a neural ranking model that directly tackles the ad-hoc retrieval task. In the same year, Lu and Li [14] proposed DeepMatch, which is a deep matching method applied to the Community-based Question Answering (CQA) and micro-blog matching tasks. Note that at the same time or even before this work, there were a number of studies focused on learning low-dimensional representations of texts with neural models [15, 16] and using them either within traditional IR models or with some new similarity metrics for ranking tasks. However, we would like to refer to those methods as representation learning models rather than neural ranking models, since they did not directly construct the ranking function with neural networks. Later, between 2014 and 2015, work on neural ranking models began to grow, such as new variants of DSSM [13], ARC I and ARC II [17], MatchPyramid [18], and so on. Most of this research focused on short text ranking tasks, such as TREC QA tracks and Microblog tracks [19]. Since 2016, the study of neural ranking models has bloomed, with significant work volume, deeper and more rigorous discussions, and much wider applications [20]. For example, researchers began to discuss the practical effectiveness of neural ranking models on different ranking tasks [21, 22]. Neural ranking models have been applied to ad-hoc retrieval [23, 24], community-based QA [25], conversational search [26], and so on. Researchers began to go beyond the architecture of neural ranking models, paying attention to new training paradigms of neural ranking models [27], alternate indexing schemes for neural representations [28], integration of

external knowledge [29, 30], and other novel uses of neural approaches for IR tasks [31, 32].

Up to now, we have seen exciting progress on neural ranking models. In academia, several neural ranking models learned from scratch can already outperform state-of-the-art LTR models with tens of hand-crafted features [33, 34]. Workshops and tutorials on this topic have attracted extensive interest in the IR community [10, 35]. Standard benchmark datasets [36, 37], evaluation tasks [38], and open-source toolkits [39] have been created to facilitate research and rigorous comparison. Meanwhile, in industry, we have also seen models such as DSSM put into a wide range of practical usage in the enterprise [40]. Neural ranking models already generate the most important features for modern search engines. However, beyond these exciting results, there is still a long way to go for neural ranking models: 1) Neural ranking models have not had the level of breakthroughs achieved by neural methods in speech recognition or computer vision; 2) There is little understanding and few guidelines on the design principles of neural ranking models; 3) We have not identified the special capabilities of neural ranking models that go beyond traditional IR models. Therefore, it is the right moment to take a look back, summarize the current status, and gain some insights for future development.

There have been some related surveys on neural approaches to IR (neural IR for short). For example, Onal et al.[20] reviewed the current landscape of neural IR research, paying attention to the application of neural methods to different IR tasks. Mitra and Craswell [41] gave an introduction to neural information retrieval. In their booklet, they talked about fundamentals of text retrieval, and briefly reviewed IR methods employing pre-trained embeddings and neural networks. In contrast to this work, this survey does not try to cover every aspect of neural IR, but will focus on and take a deep look into ranking models with deep neural networks. Specifically, we formulate the existing neural ranking models under a unified framework, and review them from different dimensions to understand their underlying assumptions, major design principles, and learning strategies. We also compare representative neural ranking models

through benchmark tasks to obtain a comprehensive empirical understanding. We hope these discussions will help researchers in neural IR learn from previous successes and failures, so that they can develop better neural ranking models in the future. In addition to the model discussion, we also introduce some trending topics in neural IR, including indexing schema, knowledge integration, visualized learning, contextual learning and model explanation. Some of these topics are important but have not been well addressed in this field, while others are very promising directions for future research.

In the following, we will first introduce some typical textual IR tasks addressed by neural ranking models in Section 2. We then provide a unified formulation of neural ranking models in Section 3. From section 4 to 6, we review the existing models with regard to different dimensions as well as making empirical comparisons between them. We discuss trending topics in Section 7 and conclude the paper in Section 8.

2. Major Applications of Neural Ranking Models

In this section, we describe several major textual IR applications where neural ranking models have been adopted and studied in the literature, including ad-hoc retrieval, question answering, community question answering, and automatic conversation. There are other applications where neural ranking models have been or could be applied, e.g., product search [12], sponsored search [42], and so on. However, due to page limitations, we will not include these tasks in this survey.

2.1. Ad-hoc Retrieval

Ad-hoc retrieval is a classic retrieval task in which the user specifies his/her information need through a query which initiates a search (executed by the information system) for documents that are likely to be relevant to the user. The term *ad-hoc* refers to the scenario where documents in the collection remain relatively static while new queries are submitted to the system continually [43].

The retrieved documents are typically returned as a ranking list through a ranking model where those at the top of the ranking are more likely to be relevant.

There has been a long research history on ad-hoc retrieval, with several well recognized characteristics and challenges associated with the task. A major characteristic of ad-hoc retrieval is the heterogeneity of the query and the documents. The query comes from a search user with potentially unclear intent and is usually very short, ranging from a few words to a few sentences [41]. The documents are typically from a different set of authors and have longer text length, ranging from multiple sentences to many paragraphs. Such heterogeneity leads to the critical vocabulary mismatch problem [44, 45]. Semantic matching, meaning matching words and phrases with similar meanings, could alleviate the problem, but exact matching is indispensable especially with rare terms [21]. Such heterogeneity also leads to diverse relevance patterns. Different hypotheses, e.g. verbosity hypothesis and scope hypothesis [46], have been proposed considering the matching of a short query against a long document. The *relevance* notion in ad-hoc retrieval is inherently vague in definition and highly user dependent, making relevance assessment a very challenging problem.

For the evaluation of different neural ranking models on the ad-hoc retrieval task, a large variety of TREC collections have been used. Specifically, retrieval experiments have been conducted over neural ranking models based on TREC collections such as Robust [21, 18], ClueWeb [21], GOV2 [33, 34] and Microblog [33], as well as logs such as the AOL log [27] and the Bing Search log [13, 47, 48, 23]. Recently, a new large scale dataset has been released, called the NTCIR WWW Task [49], which is suitable for experiments on neural ranking models.

2.2. Question Answering

Question-answering (QA) attempts to automatically answer questions posed by users in natural languages based on some information resources. The questions could be from a closed or open domain [50], while the information resources could vary from structured data (e.g., knowledge base) to unstructured

data (e.g., documents or Web pages) [51]. There have been a variety of task formats for QA, including multiple-choice selection [52], answer passage/sentence retrieval [53, 37], answer span locating [54], and answer synthesizing from multiple sources [55]. However, some of the task formats are usually not treated as an IR problem. For example, multiple-choice selection is typically formulated as a classification problem while answer span locating is usually studied under the machine reading comprehension topic. In this survey, therefore, we focus on answer passage/sentence retrieval as it can be formulated as a typical IR problem and addressed by neural ranking models. Hereafter, we will refer to this specific task as QA for simplicity.

Compared with ad-hoc retrieval, QA shows reduced heterogeneity between the question and the answer passage/sentence. On one hand, the question is usually in natural language, which is longer than keyword queries and clearer in intent description. On the other hand, the answer passages/sentences are usually much shorter text spans than documents (e.g., the answer passage length of WikiPassageQA data is about 133 words [56]), leading to more concentrated topics/semantics. However, vocabulary mismatch is still a basic problem in QA. The notion of relevance is relatively clear in QA, i.e., whether the target passage/sentence answers the question, but assessment is challenging. Ranking models need to capture the patterns expected in the answer passage/sentence based on the intent of the question, such as the matching of the context words, the existence of the expected answer type, and so on.

For the evaluation of QA tasks, several benchmark data sets have been developed, including TREC QA [53], WikiQA [37], WebAP [57, 58], InsuranceQA [59], WikiPassageQA [56] and MS MARCO [36]. A variety of neural ranking models [60, 19, 61, 25, 14] have been tested on these data sets.

2.3. Community Question Answering

Community question answering (CQA) aims to find answers to users' questions based on existing QA resources in CQA websites, such as Quora [3], Yahoo! Answers [4], Stack Overflow [5], and Zhihu [6]. As a retrieval task, CQA can be further divided into two categories. The first is to directly retrieval answers from the answer pool, which is similar to the above QA task with some additional user behavioral data (e.g., upvotes/downvotes) [62]. So we will not discuss this format here again. The second is to retrieve similar questions from the question pool, based on the assumption that answers to similar question could answer new questions. Unless otherwise noted, we will refer to the second task format as CQA.

Since it involves the retrieval of similar questions, CQA is significantly different from the previous two tasks due to the homogeneity between the input question and target question. Specifically, both input and target questions are short natural language sentences (e.g. the question length in Yahoo! Answers is between 9 and 10 words on average [63]), describing users' information needs. Relevance in CQA refers to semantic equivalence/similarity, which is clear and symmetric in the sense that the two questions are exchangeable in the relevance definition. However, vocabulary mismatch is still a challenging problem as both questions are short and there exist different expressions for the same intent.

For evaluation of the CQA task, a large variety of data sets have been released for research. The well-known data sets include the Quora Dataset[7], Yahoo! Answers Dataset [25] and SemEval-2017 Task3 [64]. The recent proposed datasets include CQADupStack[8] [65], ComQA[9] [66] and LinkSO [67]. A variety of neural ranking models [68, 18, 69, 70, 25] have been tested on these

[3] https://www.quora.com/
[4] https://answers.yahoo.com
[5] https://www.stackoverflow.com
[6] https://zhihu.com
[7] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
[8] https://github.com/D1Doris/CQADupStack
[9] http://qa.mpi-inf.mpg.de/comqa

data sets.

2.4. Automatic Conversation

Automatic conversation (AC) aims to create an automatic human-computer dialog process for the purpose of question answering, task completion, and social chat (i.e., chit-chat) [71]. In general, AC could be formulated either as an IR problem that aims to rank/select a proper response from a dialog repository [72] or a generation problem that aims to generate an appropriate response with respect to the input utterance [73]. In this paper, we restrict AC to the social chat task with the IR formulation, since question answering has already been covered in the above QA task and task completion is usually not taken as an IR problem. From the perspective of conversation context, the IR-based AC could be further divided into single-turn conversation[74] or multi-turn conversation [75].

When focusing on social chat, AC also shows homogeneity similar to CQA. That is, both the input utterance and the response are short natural language sentences (e.g., the utterance length of Ubuntu Dialog Corpus is between 10 to 11 words on average and the median conversation length of it is 6 words [76]). Relevance in AC refers to certain semantic correspondence (or coherent structure) which is broad in definition, e.g., given an input utterance "OMG I got myopia at such an 'old' age", the response could range from general (e.g., "Really?") to specific (e.g., "Yeah. Wish a pair of glasses as a gift") [26]. Therefore, vocabulary mismatch is no longer the central challenge in AC, as we can see from the example that a good response does not require semantic matching between the words. Instead, it is critical to model correspondence/coherence and avoid general trivial responses.

For the evaluation of different neural ranking models on the AC task, several conversation collections have been collected from social media such as forums, Twitter and Weibo. Specifically, experiments have been conducted over neural ranking models based on collections such as Ubuntu Dialog Corpus (UDC) [75, 77, 78], Sina Weibo dataset [74, 26, 79, 80], MSDialog [81, 30, 82] and the "campaign" NTCIR STC [83].

3. A Unified Model Formulation

Neural ranking models are mostly studied within the LTR framework. In this section, we give a unified formulation of neural ranking models from a generalized view of LTR problems.

Suppose that $\mathcal{S}$ is the *generalized* query set, which could be the set of search queries, natural language questions or input utterances, and $\mathcal{T}$ is the *generalized* document set, which could be the set of documents, answers or responses. Suppose that $\mathcal{Y} = \{1, 2, \cdots, l\}$ is the label set where labels represent grades. There exists a total order between the grades $l \succ l-1 \succ \cdots \succ 1$, where $\succ$ denotes the order relation. Let $s_i \in \mathcal{S}$ be the i-th query, $T_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,n_i}\} \in \mathcal{T}$ be the set of documents associated with the query s_i, and $\mathbf{y}_i = \{y_{i,1}, y_{i,2}, \cdots, y_{i,n_i}\}$ be the set of labels associated with query s_i, where n_i denotes the size of T_i and $\mathbf{y}_i$ and $y_{i,j}$ denotes the relevance degree of $t_{i,j}$ with respect to s_i. Let $\mathcal{F}$ be the function class and $f(s_i, t_{i,j}) \in \mathcal{F}$ be a ranking function which associates a relevance score with a query-document pair. Let $L(f; s_i, t_{i,j}, \mathbf{y}_{i,j})$ be the loss function defined on prediction of f over the query-document pair and their corresponding label. So a generalized LTR problem is to find the optimal ranking function f^* by minimizing the loss function over some labeled dataset

$$f^* = \arg\min \sum_i \sum_j L(f; s_i, t_{i,j}, y_{i,j}) \tag{1}$$

Without loss of generality, the ranking function f could be further abstracted by the following unified formulation

$$f(s, t) = g(\psi(s), \phi(t), \eta(s, t)) \tag{2}$$

where s and t are two input texts, ψ, ϕ are representation functions which extract features from s and t respectively, η is the interaction function which extracts features from (s, t) pair, and g is the evaluation function which computes the relevance score based on the feature representations.

Note that for traditional LTR approaches [3], functions ψ, ϕ and η are usually set to be fixed functions (i.e., manually defined feature functions). The

evaluation function g can be any machine learning model, such as logistic regression or gradient boosting decision tree , which could be learned from the training data. For neural ranking models, in most cases, all the functions ψ, ϕ, η and g are encoded in the network structures so that all of them can be learned from training data.

In traditional LTR approaches, the inputs s and t are usually raw texts. In neural ranking models, we consider that the inputs could be either raw texts or word embeddings. In other words, embedding mapping is considered as a basic input layer, not included in ψ, ϕ and η.

4. Model Architecture

Based on the above unified formulation, here we review existing neural ranking model architectures to better understand their basic assumptions and design principles.

4.1. Symmetric vs. Asymmetric Architectures

Starting from different underlying assumptions over the input texts s and t, two major architectures emerge in neural ranking models, namely symmetric architecture and asymmetric architecture.

Symmetric Architecture: The inputs s and t are assumed to be homogeneous, so that symmetric network structure could be applied over the inputs. Note here symmetric structure means that the inputs s and t can exchange their positions in the input layer without affecting the final output. Specifically, there are two representative symmetric structures, namely siamese networks and symmetric interaction networks.

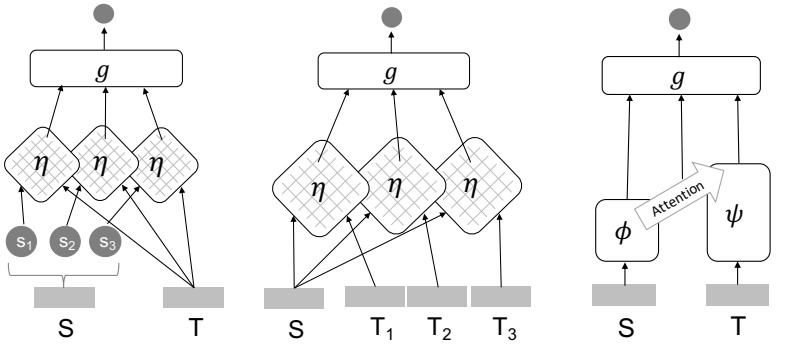
Siamese networks literally imply symmetric structure in the network architecture. Representative models include DSSM [13], CLSM [47] and LSTM-RNN [48]. For example, DSSM represents two input texts with a unified process including the letter-trigram mapping followed by the multi-layer perceptron (MLP) transformation, i.e., function ϕ is the same as function ψ. After that a

cosine similarity function is applied to evaluate the similarity between the two representations, i.e., function g is symmetric. Similarly, CLSM [47] replaces the representation functions ψ and ϕ by two identical convolutional neural networks (CNNs) in order to capture the local word order information. LSTM-RNN [48] replaces ψ and ϕ by two identical long short-term memory (LSTM) networks in order to capture the long-term dependence between words.

Symmetric interaction networks, as shown by the name, employ a symmetric interaction function to represent the inputs. Representative models include DeepMatch [14], Arc-II [17], MatchPyramid [18] and Match-SRNN [69]. For example, Arc-II defines an interaction function η over s and t by computing similarity (i.e., weighted sum) between every n-gram pair from s and t, which is symmetric in nature. After that, several convolutional and max-pooling layers are leveraged to obtain the final relevance score, which is also symmetric over s and t. MatchPyramid defines a symmetric interaction function η between every word pair from s and t to capture fine-grained interaction signals. It then leverages a symmetric evaluation function g, i.e., several 2D CNNs and a dynamic pooling layer, to produce the relevance score. A similar process can be found in DeepMatch and Match-SRNN.

Symmetric architectures, with the underlying homogeneous assumption, can fit well with the CQA and AC tasks, where s and t usually have similar lengths and similar forms (i.e., both are natural language sentences). They may sometimes work for the ad-hoc retrieval or QA tasks if one only uses document titles/snippets [13] or short answer sentences [61] to reduce the heterogeneity between the two inputs.

Asymmetric Architecture: The inputs s and t are assumed to be heterogeneous, so that asymmetric network structures should be applied over the inputs. Note here asymmetric structure means if we change the position of the inputs s and t in the input layer, we will obtain totally different output. Asymmetric architectures have been introduced mainly in the ad-hoc retrieval task [13, 33], due to the inherent heterogeneity between the query and the document as discussed in Section 2.1. Such structures may also work for the QA task



(a) Query Split (b) Document Split (c) One-way Attention

Figure 1: Three types of Asymmetric Architecture.

where answer passages are ranked against natural language questions [84].

Here we take the ad-hoc retrieval scenario as an example to analyze the asymmetric architecture. We find there are three major strategies used in the asymmetric architecture to handle the heterogeneity between the query and the document, namely query split, document split, and joint split.

- *Query split* is based on the assumption that most queries in ad-hoc retrieval are keyword based, so that we can split the query into terms to match against the document, as illustrated in Figure 1(a). A typical model based on this strategy is DRMM [21]. DRMM splits the query into terms and defines the interaction function η as the matching histogram mapping between each query term and the document. The evaluation function g consists of two parts, i.e., a feed-forward network for term-level relevance computation and a gating network for score aggregation. Obviously such a process is asymmetric with respect to the query and the document. K-NRM [85] also belongs to this type of approach. It introduces a kernel pooling function to approximate matching histogram mapping to enable end-to-end learning.

- *Document split* is based on the assumption that a long document could be partially relevant to a query under the scope hypothesis [2], so that we

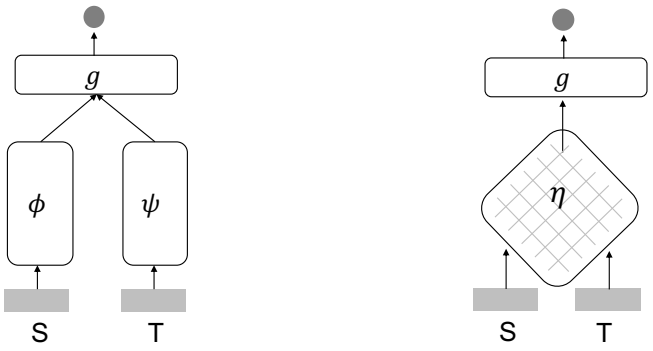
split the document to capture fine-grained interaction signals rather than treat it as a whole, as depicted in Figure 1(b). A representative model based on this strategy is HiNT [34]. In HiNT, the document is first split into passages using a sliding window. The interaction function η is defined as the cosine similarity and exact matching between the query and each passage. The evaluation function g includes the local matching layers and global decision layers.

- *Joint split*, by its name, uses both assumptions of query split and document split. A typical model based on this strategy is DeepRank [33]. Specifically, DeepRank splits the document into term-centric contexts with respect to each query term. It then defines the interaction function η between the query and term-centric contexts in several ways. The evaluation function g includes three parts, i.e., term-level computation, term-level aggregation, and global aggregation. Similarly, PACRR [24] takes the query as a set of terms and splits the document using the sliding window as well as the first-k term window.

In addition, in neural ranking models applied for QA, there is another popular strategy leading the asymmetric architecture. We name it *one-way attention mechanism* which typically leverages the question representation to obtain the attention over candidate answer words in order the enhance the answer representation, as illustrated in Figure 1(c). For example, IARNN [86] and CompAgg [87] get the attentive answer representation sequence that weighted by the question sentence representation.

4.2. Representation-focused vs. Interaction-focused Architectures

Based on different assumptions over the features (extracted by the representation function ϕ, ψ or the interaction function η) for relevance evaluation, we can divide the existing neural ranking models into another two categories of architectures, namely representation-focused architecture and interaction-focused architecture, as illustrated in Figure 2. Besides these two basic categories, some



(a) Representation-focused (b) Interaction-focused

Figure 2: Representation-focused and Interaction-focused Architectures.

neural ranking models adopt a hybrid way to enjoy the merits of both architectures in learning relevance features.

Representation-focused Architecture: The underlying assumption of this type of architecture is that relevance depends on compositional meaning of the input texts. Therefore, models in this category usually define complex representation functions ϕ and ψ (i.e., deep neural networks), but no interaction function η, to obtain high-level representations of the inputs s and t, and uses some simple evaluation function g (e.g. cosine function or MLP) to produce the final relevance score. Different deep network structures have been applied for ϕ and ψ, including fully-connected networks, convolutional networks and recurrent networks.

- To our best knowledge, DSSM [13] is the only one that uses the fully-connected network for the functions ϕ and ψ, which has been described in Section 4.1.

- Convolutional networks have been used for ϕ and ψ in Arc-I [17], CNTN [25] and CLSM [47]. Take Arc-I as an example, stacked 1D convolutional layers and max pooling layers are applied on the input texts s and t to produce their high-level representations respectively. Arc-I then concatenates the two representations and applies an MLP as the evaluation function g.

The main difference between CNTN and Arc-I is the function g, where the neural tensor layer is used instead of the MLP. The description on CLSM could be found in Section 4.1.

- Recurrent networks have been used for ϕ and ψ in LSTM-RNN [48] and MV-LSTM [88]. LSTM-RNN uses a one-directional LSTM as ϕ and ψ to encode the input texts, which has been described in Section 4.1. MV-LSTM employs a bi-directional LSTM instead to encode the input texts. Then, the top-k strong matching signals between the two high-level representations are fed to an MLP to generate the relevance score.

By evaluating relevance based on high-level representations of each input text, representation-focused architecture better fits tasks with the global matching requirement [21]. This architecture is also more suitable for tasks with short input texts (since it is often difficult to obtain good high-level representations of long texts). Tasks with these characteristics include CQA and AC as shown in Section 2. Moreover, models in this category are efficient for online computation, since one can pre-calculate representations of the texts offline once ϕ and ψ have been learned.

Interaction-focused Architecture: The underlying assumption of this type of architecture is that relevance is in essence about the relation between the input texts, so it would be more effective to directly learn from interactions rather than from individual representations. Models in this category thus define the interaction function η rather than the representation functions ϕ and ψ, and use some complex evaluation function g (i.e., deep neural networks) to abstract the interaction and produce the relevance score. Different interaction functions have been proposed in literature, which could be divided into two categories, namely non-parametric interaction functions and parametric interaction functions.

- *Non-parametric interaction functions* are functions that reflect the closeness or distance between inputs without learnable parameters. In this

category, some are defined over each pair of input word vectors, such as binary indicator function [18, 33], cosine similarity function [18, 61, 33], dot-product function [18, 33, 34] and radial-basis function [18]. The others are defined between a word vector and a set of word vectors, e.g. the matching histogram mapping in DRMM [21] and the kernel pooling layer in K-NRM [85].

- *Parametric interaction functions* are adopted to learn the similarity/distance function from data. For example, Arc-II [17] uses 1D convolutional layer for the interaction bwteen two phrases. Match-SRNN [69] introduces the neural tensor layer to model complex interactions between input words. Some BERT-based model [89] takes attention as the interaction function to learn the interaction vector (i.e., [CLS] vector) between inputs. In general, parametric interaction functions are adopted when there is sufficient training data since they bring the model flexibility at the expense of larger model complexity.

By evaluating relevance directly based on interactions, the interaction-focused architecture can fit most IR tasks in general. Moreover, by using detailed interaction signals rather than high-level representations of individual texts, this architecture could better fit tasks that call for specific matching patterns (e.g., exact word matching) and diverse matching requirement [21], e.g., ad-hoc retrieval. This architecture also better fit tasks with heterogeneous inputs, e.g., ad-hoc retrieval and QA, since it circumvents the difficulty of encoding long texts. Unfortunately, models in this category are not efficient for online computation as previous representation-focused models, since the interaction function η cannot be pre-calculated until we see the input pair (s, t). Therefore, a better way for practical usage is to apply these two types of models in a "telescope" setting, where representation-focused models could be applied in an early search stage while interaction-focused models could be applied later on.

It is worth noting that parts of the interaction-focused architectures have some connections to those in the computer vision (CV) area. For example, the

designs of MatchPyramid [18] and PACRR [24] are inspired by the neural models for the image recognition task. By viewing the matching matrix as a 2-D image, a CNN network is naturally applied to extract hierarchical matching patterns for relevance estimation. These connections indicate that although neural ranking models are mostly applied over textual data, one may still borrow many useful ideas in neural architecture design from other domains.

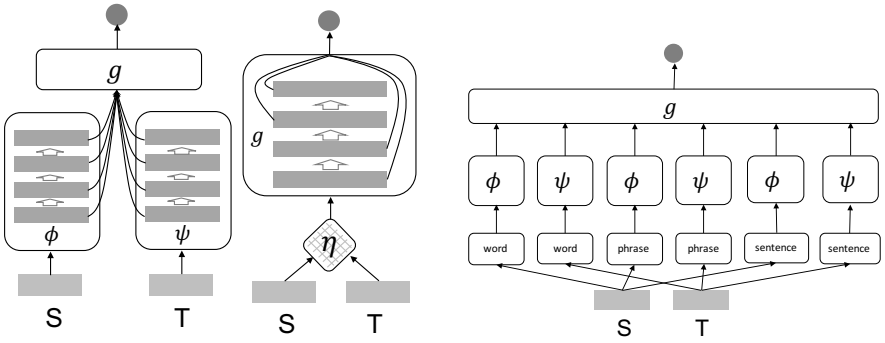
Hybrid Architecture: In order to take advantage of both representation-focused and interaction-focused architectures, a natural way is to adopt a hybrid architecture for feature learning. We find that there are two major hybrid strategies to integrate the two architectures, namely combined strategy and coupled strategy.

- Combined strategy is a loose hybrid strategy, which simply adopts both representation-focused and interaction-focused architectures as sub-models and combines their outputs for final relevance estimation. A representative model using this strategy is DUET [23]. DUET employs a CLSM-like architecture (i.e., a distributed network) and a MatchPyramid-like architecture (i.e., a local network) as two sub-models, and uses a sum operation to combine the scores from the two networks to produce the final relevance score.

- Coupled strategy, on the other hand, is a compact hybrid strategy. A typical way is to learn representations with attention across the two inputs. Therefore, the representation functions ϕ and ψ and the interaction function η are compactly integrated. Representative models using this strategy include IARNN [86] and CompAgg [87], which have been discussed in the Section 4.1. Both models learn the question and answer representations via some one-way attention mechanism.

4.3. Single-granularity vs. Multi-granularity Architecture

The final relevance score is produced by the evaluation function g, which takes the features from ϕ, ψ, and η as input for estimation. Based on different



(a) Vertical Multi-granularity (b) Horizontal Multi-granularity

Figure 3: Multi-granularity Architectures.

assumptions on the estimation process for relevance, we can divide existing neural ranking models into two categories, namely single-granularity models and multi-granularity models.

Single-granularity Architecture: The underlying assumption of the single-granularity architecture is that relevance can be evaluated based on the high-level features extracted by ϕ, ψ and η from the single-form text inputs. Under this assumption, the representation functions ϕ, ψ and the interaction function η are actually viewed as black-boxes to the evaluation function g. Therefore, g only takes their final outputs for relevance computation. Meanwhile, the inputs s and t are simply viewed a set/sequence of words or word embeddings without any additional language structures.

Obviously, the assumption underlying the single-granularity architecture is very simple and basic. Many neural ranking models fall in this category, with either symmetric (e.g., DSSM and MatchPyramid) or asymmetric (e.g., DRMM and HiNT) architectures, either representation-focused (e.g., ARC-I and MV-LSTM) or interaction-focused (e.g., K-NRM and Match-SRNN).

Multi-granularity Architecture: The underlying assumption of the multi-granularity architecture is that relevance estimation requires multiple granularities of features, either from different-level feature abstraction or based on different types of language units of the inputs. Under this assumption, the represen-

tation functions ϕ, ψ and the interaction function η are no longer black-boxes to g, and we consider the language structures in s and t. We can identify two basic types of multi-granularity, namely vertical multi-granularity and horizontal multi-granularity, as illustrated in Figure 3.

- *Vertical multi-granularity* takes advantage of the hierarchical nature of deep networks so that the evaluation function g could leverage different-level abstraction of features for relevance estimation. For example, In MultigranCNN [90], the representation functions ψ and ϕ are defined as two CNN networks to encode the input texts respectively, and the evaluation function g takes the output of each layer for relevance estimation. MACM [91] builds a CNN over the interaction matrix from η, uses MLP to generate a layer-wise score for each abstraction level of the CNN, and aggregates all the layers' scores for the final relevance estimation. Similar ideas can also be found in MP-HCNN [92] and MultiMatch [93].

- *Horizontal multi-granularity* is based on the assumption that language has intrinsic structures (e.g., phrases or sentences), and we shall consider different types of language units, rather than simple words, as inputs for better relevance estimation. Models in this category typically enhance the inputs by extending it from words to phrases/n-grams or sentences, apply certain single-granularity architectures over each input form, and aggregate all the granularity for final relevance output. For example, in [94], a CNN and an LSTM are applied to obtain the character-level, word-level, and sentence-level representations of the inputs, and each level representations are then interacted and aggregated by the evaluation function g to produce the final relevance score. Similar ideas can be found in Conv-KNRM [84] and MIX [95].

As we can see, the multi-granularity architecture is a natural extension of the single-granularity architecture, which takes into account the inherent language structures and network structures for enhanced relevance estimation. With

multi-granularity features extracted, models in this category are expected to better fit tasks that require fine-grained matching signals for relevance computation, e.g., ad-hoc retrieval [84] and QA [95]. However, the enhanced model capability is often reached at the expense of larger model complexity.

5. Model Learning

Beyond the architecture, in this section, we review the major learning objectives and training strategies adopted by neural ranking models for comprehensive understadning.

5.1. Learning objective

Similar to other LTR algorithms, the learning objective of neural ranking models can be broadly categorized into three groups: *pointwise*, *pairwise*, and *listwise*. In this section, we introduce a couple of popular ranking loss functions in each group, and discuss their unique advantages and disadvantages for the applications of neural ranking models in different IR tasks.

5.1.1. Pointwise Ranking Objective

The idea of pointwise ranking objectives is to simplify a ranking problem to a set of classification or regression problems. Specifically, given a set of query-document pairs $(s_i, t_{i,j})$ and their corresponding relevance annotation $y_{i,j}$, a pointwise learning objective tries to optimize a ranking model by requiring it to directly predict $y_{i,j}$ for $(s_i, t_{i,j})$. In other words, the loss functions of pointwise learning objectives are computed based on each (s, t) pair independently. This can be formulated as

$$L(f; \mathcal{S}, \mathcal{T}, \mathcal{Y}) = \sum_i \sum_j L(y_{i,j}, f(s_i, t_{i,j})) \tag{3}$$

For example, one of the most popular pointwise loss functions used in neural ranking models is *Cross Entropy*:

$$L(f; \mathcal{S}, \mathcal{T}, \mathcal{Y}) = -\sum_i \sum_j y_{i,j} \log(f(s_i, t_{i,j})) + (1 - y_{i,j}) \log(1 - f(s_i, t_{i,j})) \tag{4}$$

where $y_{i,j}$ is a binary label or annotation with probabilistic meanings (e.g., clickthrough rate), and $f(s_i, t_{i,j})$ needs to be rescaled into the range of 0 to 1 (e.g., with a sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$). Example applications include the Convolutional Neural Network for question answering [19]. There are other pointwise loss functions such as *Mean Squared Error* for numerical labels, but they are more commonly used in recommendation tasks.

The advantages of pointwise ranking objectives are two-fold. First, pointwise ranking objectives are computed based on each query-document pair $(s_i, t_{i,j})$ separately, which makes it simple and easy to scale. Second, the outputs of neural models learned with pointwise loss functions often have real meanings and value in practice. For instance, in sponsored search, a model learned with cross entropy loss and clickthrough rates can directly predict the probability of user clicks on search ads, which is more important than creating a good result list in some application scenarios.

In general, however, pointwise ranking objectives are considered to be less effective in ranking tasks. Because pointwise loss functions consider no document preference or order information, they do not guarantee to produce the best ranking list when the model loss reaches the global minimum. Therefore, better ranking paradigms that directly optimize document ranking based on pairwise loss functions and listwise loss functions have been proposed for LTR problems.

5.1.2. Pairwise Ranking Objective

Pairwise ranking objectives focus on optimizing the relative preferences between documents rather than their labels. In contrast to pointwise methods where the final ranking loss is the sum of loss on each document, pairwise loss functions are computed based on the permutations of all possible document pairs [96]. It usually can be formalized as

$$L(f; \mathcal{S}, \mathcal{T}, \mathcal{Y}) = \sum_{i} \sum_{(j,k), y_{i,j} \succ y_{i,k}} L(f(s_i, t_{i,j}) - f(s_i, t_{i,k})) \tag{5}$$

where $t_{i,j}$ and $t_{i,k}$ are two documents for query s_i and $t_{i,j}$ is preferable comparing to $t_{i,k}$ (i.e., $y_{i,j} \succ y_{i,k}$). For instance, a well-known pairwise loss function is

Hinge loss:

$$L(f;\mathcal{S},\mathcal{T},\mathcal{Y}) = \sum_{i} \sum_{(j,k),y_{i,j}\succ y_{i,k}} \max(0, 1 - f(s_i, t_{i,j}) + f(s_i, t_{i,k})) \tag{6}$$

Hinge loss has been widely used in the training of neural ranking models such as DRMM [21] and K-NRM [85]. Another popular pairwise loss function is the pairwise cross entropy defined as

$$L(f;\mathcal{S},\mathcal{T},\mathcal{Y}) = -\sum_{i} \sum_{(j,k),y_{i,j}\succ y_{i,k}} \log \sigma(f(s_i, t_{i,j}) - f(s_i, t_{i,k})) \tag{7}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$. Pairwise cross entropy is first proposed in RankNet by Burges et al. [97], which is considered to be one of the initial studies on applying neural network techniques to ranking problems.

Ideally, when pairwise ranking loss is minimized, all preference relationships between documents should be satisfied and the model will produce the optimal result list for each query. This makes pairwise ranking objectives effective in many tasks where performance is evaluated based on the ranking of relevant documents. In practice, however, optimizing document preferences in pairwise methods does not always lead to the improvement of final ranking metrics due to two reasons: (1) it is impossible to develop a ranking model that can correctly predict document preferences in all cases; and (2) in the computation of most existing ranking metrics, not all document pairs are equally important. This means that the performance of pairwise preference prediction is not equal to the performance of the final retrieval results as a list. Given this problem, previous studies [98, 99, 100, 101] further proposed listwise ranking objectives for learning to rank.

5.1.3. Listwise Ranking Objective

The idea of listwise ranking objectives is to construct loss functions that directly reflect the model's final performance in ranking. Instead of comparing two documents each time, listwise loss functions compute ranking loss with each query and their candidate document list together. Formally, most existing

listwise loss functions can be formulated as

$$L(f; \mathcal{S}, \mathcal{T}, \mathcal{Y}) = \sum_i L(\{y_{i,j}, f(s_i, t_{i,j}) | t_{i,j} \in \mathcal{T}_i\}) \tag{8}$$

where $\mathcal{T}_i$ is the set of candidate documents for query s_i. Usually, L is defined as a function over the list of documents sorted by $y_{i,j}$, which we refer to as π_i, and the list of documents sorted by $f(s_i, t_{i,j})$. For example, Xia et al. [98] proposed *ListMLE* for listwise ranking as

$$L(f; \mathcal{S}, \mathcal{T}, \mathcal{Y}) = \sum_i \sum_{j=1}^{|\pi_i|} \log P(y_{i,j} | \mathcal{T}_i^{(j)}, f) \tag{9}$$

where $P(y_{i,j} | \mathcal{T}_i^{(j)}, f)$ is the probability of selecting the jth document in the optimal ranked list π_i with f:

$$P(y_{i,j} | \mathcal{T}_i^{(j)}, f) = \frac{\exp(f(s_i, t_{i,j}))}{\sum_{k=j}^{|\pi_i|} \exp(f(s_i, t_{i,k}))} \tag{10}$$

Intuitively, ListMLE is the log likelihood of the optimal ranked list given the current ranking function f, but computing log likelihood on all the result positions is computationally prohibitive in practice. Thus, many alternative functions have been proposed for listwise ranking objectives in the past ten years. One example is the *Attention Rank* function used in the Deep Listwise Context Model proposed by Ai et al. [101]:

$$\begin{aligned} L(f; \mathcal{S}, \mathcal{T}, \mathcal{Y}) &= -\sum_i \sum_j P(t_{i,j} | \mathcal{Y}_i, \mathcal{T}_i) \log P(t_{i,j} | f, \mathcal{T}_i) \\ \text{where } P(t_{i,j} | \mathcal{Y}_i, \mathcal{T}_i) &= \frac{\exp(y_{i,j})}{\sum_{k=1}^{|\mathcal{T}_i|} \exp(y_{i,k})}, \\ P(t_{i,j} | fi, \mathcal{T}_i) &= \frac{\exp(f(s_i, t_{i,j}))}{\sum_{k=1}^{|\mathcal{T}_i|} \exp(f(s_i, t_{i,k}))} \end{aligned} \tag{11}$$

When the labels of documents (i.e., $y_{i,j}$) are binary, we can further simplify the Attention Rank function with a softmax cross entropy function as

$$L(f; \mathcal{S}, \mathcal{T}, \mathcal{Y}) = -\sum_i \sum_j y_{i,j} \log \frac{\exp(f(s_i, t_{i,j}))}{\sum_{k=1}^{|\mathcal{T}_i|} \exp(f(s_i, t_{i,k}))} \tag{12}$$

The softmax-based listwise ranking loss is one of the most popular learning objectives for neural ranking models such as GSF [102]. It is particularly useful

when we train neural ranking models with user behavior data (e.g., clicks) under the unbiased learning framework [103]. There are other types of listwise loss functions proposed under different ranking frameworks in the literature [100, 99]. We ignore them in this paper since they are not popular in the studies of neural IR.

While listwise ranking objectives are generally more effective than pairwise ranking objectives, their high computational cost often limits their applications. They are suitable for the re-ranking phase over a small set of candidate documents. Since many practical search systems now use neural models for document re-ranking, listwise ranking objectives have become increasingly popular in neural ranking frameworks [13, 47, 23, 101, 102, 103].

5.1.4. Multi-task Learning Objective

In some cases, the optimization of neural ranking models may include the learning of multiple ranking or non-ranking objectives at the same time. The motivation behind this approach is to use the information from one domain to help the understanding of information from other domains. For example, Liu et al. [104] proposed to unify the representation learning process for query classification and Web search by training a deep neural network in which the final layer of hidden variables are used to optimize both a classification loss and a ranking loss. Chapelle et al. [105] proposed a multi-boost algorithm to simultaneously learn ranking functions based on search data collected from 15 countries.

In general, the most common methodology used by existing multi-task learning algorithms is to construct shared representations that are universally effective for ranking in multiple tasks or domains. To do so, previous studies mostly focus on constructing regularizations or restrictions on model optimizations so that the final model is not specifically designed for a single ranking objective [104, 105]. Inspired by recent advances on generative adversarial networks (GAN) [106], Cohen et al. [107] introduced an adversarial learning framework that jointly learns a ranking function with a discriminator which can distin-

guish data from different domains. By training the ranking function to produce representations that cannot be discriminated by the discriminator, they teach the ranking system to capture domain-independent patterns that are usable in cross-domain applications. This is important as it can significantly alleviate the problem of data sparsity in specific tasks and domains.

5.2. Training Strategies

Given the data available for training a neural ranking model, an appropriate training strategy should be chosen. In this section, we briefly review a set of effective training strategies for neural ranking models, including supervised, semi-supervised, and weakly supervised learning.

Supervised learning refers to the most common learning strategy in which query-document pairs are labeled. The data can be labeled by expert assessors, crowdsourcing, or can be collected from the user interactions with a search engine as implicit feedback. In this training strategy, it is assumed that a sufficient amount of labeled training data is available. Given this training strategy, one can train the model using any of the aforementioned learning objectives, e.g., pointwise and pairwise. However, since neural ranking models are usually data "hungry", academic researchers can only learn models with constrained parameter spaces under this training paradigm due to the limited annotated data. This has motivated researchers to study learning from limited data for information retrieval [108].

Weakly supervised learning refers to a learning strategy in which the query-document labels are automatically generated using an existing retrieval model, such as BM25. The use of pseudo-labels for training ranking models has been proposed by Asadi et al. [109]. More recently, Dehghani et al. [27] proposed to train neural ranking models using weak supervision and observed up to 35% improvement compared to BM25 which plays the role of weak labeler. This learning strategy does not require labeled training data. In addition to ranking, weak supervision has shown successful results in other information retrieval tasks, including query performance prediction [110], learning relevance-based

word embedding [111], and efficient learning to rank [112].

Semi-supervised learning refers to a learning strategy that leverages a small set of labeled query-document pairs plus a large set of unlabeled data. Semi-supervised learning has been extensively studied in the context of learning to rank. Preference regularization [113], feature extraction using KernelPCA [114], and pseudo-label generation using labeled data [115] are examples of such approaches. In the realm of neural models, fine-tuning weak supervision models using a small set of labeled data [27] and controlling the learning rate in learning from weakly supervised data using a small set of labeled data [116] are another example of semi-supervised approaches to ranking. Recently, Li et al. [117] proposed a neural model with a joint supervised and unsupervised loss functions. The supervised loss accounts for the error in query-document matching, while the unsupervised loss computes the document reconstruction error (i.e., auto-encoders).

6. Model Comparison

In this section, we compare the empirical evaluation results of the previously reviewed neural ranking models on several popular benchmark data sets. We mainly survey and analyze the published results of neural ranking models for the ad-hoc retrieval and QA tasks. Note that sometimes it is difficult to compare published results across different papers - small changes such as different tokenization, stemming, etc. can lead to significant differences. Therefore, we attempt to collect results from papers that contain comparisons across some of these models performed at a single site for fairness .

6.1. Empirical Comparison on Ad-hoc Retrieval

To better understand the performances of different neural ranking models on ad-hoc retrieval, we show the published experimental results on benchmark datasets. Here, we choose three representative datasets for ad-hoc retrieval: (1) Robust04 dataset is a standard ad-hoc retrieval dataset where the queries

are from TREC Robust Track 2004. (2) Gov2_{MQ2007} is an Web Track ad-hoc retrieval dataset where the collection is the Gov2 corpus. The queries are from the Million Query Track of TREC 2007. (3) Sougou-Log dataset [85] is built on query logs sampled from search logs of Sougou.com. (4) WT09-14 is the 2009-2014 TREC Web Track, which are based on the ClueWeb09 and ClueWeb12 datasets. The detailed data statistics can be found in related literature [21, 33, 34, 85, 118].

For meaningful comparison, we have tried our best to restrict the reported results to be under the same experimental settings. Specifically, experiments on Robust04 take the title as the query, and all the documents are processed with the Galago Search Engine[10] [21, 28]. For experiments on the Gov2_{MQ2007} dataset, all the queries and documents are processed using the Galago Search Engine under the same setting as described in [33, 34]. Besides, the results on the WT09-14 dataset and the Sougou-Log dataset are all from a same paper [118, 84] respectively.

Table 1 shows an overview of previous published results on ad-hoc retrieval datasets. We have included some well-known probabilistic retrieval models, pseudo-relevance feedback (PRF) models and LTR models as baselines. Based on the results, we have the following observations:

1. The probabilistic models (i.e., QL and BM25), although simple, can already achieve reasonably good performance. The traditional PRF model (i.e., RM3) and LTR models (i.e., RankSVM and LambdaMart) with human designed features are strong baselines whose performance is hard to beat for most neural ranking models based on raw texts. However, the PRF technique can also be leveraged to enhance neural ranking models (e.g., SNRM+PRF [28] and NPRF+DRMM [119] in Table 1), while human designed LTR features can be integrated into neural ranking models [33, 31] to improve the ranking performance.

[10] http://www.lemurproject.org/galago.php

Table 1: Overview of previously published results on ad hoc retrieval datasets. The citation in each row denotes the original paper where the method is proposed. The superscripts 1-6 denote that the results are cited from [21],[33],[34],[118], [28], [119], [84] respectively. The subscripts denote the model architecture belongs to (S)ymmetric or (A)symmetric/(R)epresentation-focused or (I)nteraction-focused or (H)ybrid/Singe-(G)ranularity or (M)ulti-granularity. The back slash symbols denote that there are no published results for the specific model on the specific data set in the related literature.

Data Set / Model	Robust04		$GOV2_{MQ2007}$		WT09-14	Sougo-Log
	MAP	P@20	MAP	P@10	ERR@20	NDCG@1
BM25[46] $(1994)^{1,2}$	0.255	0.370	0.450	0.366	\	0.142
QL[120] $(1998)^{1,4}$	0.253	0.369	\	\	0.113	0.126
RM3[121](2001)^{5}	0.287	0.377	\	\	\	\
RankSVM[122] $(2002)^{2}$	\	\	0.464	0.381	\	0.146
LambdaMart[100] $(2010)^{2}$	\	\	0.468	0.384	\	\
DSSM[13] $(2013)^{1,2}_{S/R/G}$	0.095	0.171	0.409	0.352	\	\
CDSSM[47] $(2014)^{1,2}_{S/R/G}$	0.067	0.125	0.364	0.291	\	0.144
ARC-I[17] $(2014)^{1,2}_{S/R/G}$	0.041	0.065	0.417	0.364	\	\
ARC-II[17] $(2014)^{1,2}_{S/I/G}$	0.067	0.128	0.421	0.366	\	\
MP[18] $(2016)^{1,2,4}_{S/I/G}$	0.189	0.290	0.434	0.371	0.148	0.218
Match-SRNN[69] $(2016)^{2}_{S/H/G}$	\	\	0.456	0.384	\	\
DRMM[21] $(2016)^{1,2,4}_{A/I/G}$	0.279	0.382	0.467	0.388	0.171	0.137
Duet[23] $(2017)^{3,4}_{A/H/G}$	\	\	0.474	0.398	0.134	\
DeepRank[33] $(2017)^{2}_{A/I/G}$	\	\	0.497	0.412	\	\
K-NRM[85] $(2017)^{4}_{A/I/G}$	\	\	\	\	0.154	0.264
PACRR[123] $(2017)^{6,4}_{A/I/M}$	0.254	0.363	\	\	0.191	\
Co-PACRR[118] $(2018)^{4}_{A/I/M}$	\	\	\	\	0.201	\
SNRM[28] $(2018)^{5}_{S/R/G}$	0.286	0.377	\	\	\	\
SNRM+PRF[28] $(2018)^{5}_{S/R/G}$	0.297	0.395	\	\	\	\
CONV-KNRM[84] $(2018)^{4}_{A/I/M}$	\	\	\	\	\	0.336
NPRF-KNRM[119] $(2018)^{6}_{A/I/G}$	0.285	0.393	\	\	\	\
NPRF-DRMM[119] $(2018)^{6}_{A/I/G}$	0.290	0.406	\	\	\	\
HiNT[34] $(2018)^{3}_{A/I/G}$	\	\	0.502	0.418	\	\

2. There seems to be a paradigm shift of the neural ranking model architectures from symmetric to asymmetric and from representation-focused to interaction-focused over time. This is consistent with our previous analysis where asymmetric and interaction-focused structures may fit better with the ad-hoc retrieval task which shows heterogeneity inherently.
3. With bigger data size in terms of distinct number of queries and labels (i.e., Sogou-Log $\succ$ GOV2$_{MQ2007}$ $\succ$ WT09-14 $\succ$ Robust04), neural models are more likely to achieve larger performance improvement against non-neural models. As we can see, the best neural models based on raw texts can significantly outperform LTR models with human designed features on Sogou-Log dataset.
4. Based on the reported results, in general, we observe that the asymmetric, interaction-focused, multi-granularity architecture can work better than the symmetric, representation-focused, single-granularity architecture on the ad-hoc retrieval tasks. There is one exception, i.e., SNRM on Robust04. However, this model was trained with a large amount of data using the weak supervision strategy, and may not be appropriate to directly compare with those models trained on Robust04 alone.

6.2. Empirical Comparison on QA

In order to understand the performance of different neural ranking models reviewed in this paper for the QA task, we survey the previously published results on three QA data sets, including TREC QA [124], WikiQA [37] and Yahoo! Answers [88]. TREC QA and WikiQA are answer sentence selection/retrieval data sets and they mainly contain factoid questions, while Yahoo! Answers is an answer passage retrieval data set sampled from the CQA website Yahoo! Answers. The detailed data statistics can be found in related literature [125, 37, 88].

We have tried our best to report results under the same experimental settings for fair comparison between different methods. Specifically, the results on

TREC QA are over the raw version of the data [126][11]. WikiQA only has a single version with the same train/ valid/ test data partitions [37]. Yahoo Answers data is the processed version from the same related work [88]. Therefore, questions and answer candidates in all the train/valid/test sets used in different surveyed papers are the same, and the results are comparable with each other.

Table 2 shows the overview of the published results on the QA benchmark data sets. We include several traditional non-neural methods as baselines. We summarize our observations as follows:

1. Unlike ad-hoc retrieval, symmetric architectures have been more widely adopted in the QA tasks possibly due to the increased homogeneity between the question and the answer, especially for answer sentence retrieval data sets like TREC QA and WikiQA.
2. Representation-focused architectures have been more adopted on short answer sentence retrieval data sets, i.e., TREC QA and WikiQA, while interaction-focused architectures have been more adopted on longer answer passage retrieval data sets, e.g., Yahoo! Answer. However, unlike ad-hoc retrieval, there seems to be no clear winner between the representation-focused architecture and the interaction-focused architecture on QA tasks.
3. Similar to ad-hoc retrieval, neural models are more likely to achieve larger performance improvement against non-neural models on bigger data sets. For example, on small data set like TREC QA, feature engineering based methods such as LCLR can achieve very strong performance. However, on large data set like WikiQA and Yahoo! Answers, we can see a clear gap between neural models and non-neural models.
4. The performance in general increases over time, which might be due to the increased model capacity as well as the adoption of some advanced approaches, e.g., the attention mechanism. For example, IARNN utilizes attention-based RNN models with GRU to get an attentive sentence representation. MIX extracts grammar information and integrates attention

[11] https://aclweb.org/aclwiki/Question_Answering_(State_of_the_art)

Table 2: Overview of previously published results on QA benchmark data sets. The citation in each row denotes the original paper where the method is proposed. The superscripts 1-10 denote that the results are cited from [37], [69], [61], [86], [88], [127], [87], [128], [125], [95] respectively. The subscripts denote the model architecture belongs to (S)ymmetric or (A)symmetric/(R)epresentation-focused or (I)nteraction-focused or (H)ybrid/Single-(G)ranularity or (M)ulti-granularity. The back slash symbols denote that there are no published results for the specific model on the specific data set in the related literature.

Data Set	TREC QA		WikiQA		Yahoo! Answers	
Model	MAP	MRR	MAP	MRR	P@1	MRR
BM25[46] (1994)^{2}	\	\	\	\	0.579	0.726
LCLR[129] (2013)^{1,9}	0.709	0.770	0.599	0.609	\	\
Word Cnt[125] (2014)^{1,9}	0.571	0.627	0.489	0.492	\	\
Wgt Word Cnt[125] (2014)^{1,9}	0.596	0.652	0.510	0.513	\	\
DeepMatch[14] (2013)$^{5}_{S/I/G}$	\	\	\	\	0.452	0.679
CNN[125] (2014)$^{1,9}_{S/R/G}$	0.569	0.661	0.619	0.628	\	\
CNN-Cnt[125] (2014)$^{1,9}_{S/R/G}$	0.711	0.785	0.652	0.665	\	\
ARC-I[17] (2014)$^{2}_{S/R/G}$	\	\	\	\	0.581	0.756
ARC-II[17] (2014)$^{2}_{S/I/G}$	\	\	\	\	0.591	0.765
CDNN[19] (2015)$^{3}_{S/R/G}$	0.746	0.808	\	\	\	\
BLSTM[60] (2015)$^{3}_{S/R/G}$	0.713	0.791	\	\	\	\
CNTN[25] (2015)$^{2,6}_{S/R/G}$	0.728	0.783	\	\	0.626	0.781
MultiGranCNN[90] (2015)$^{2}_{S/I/M}$	\	\	\	\	0.725	0.840
LSTM-RNN[48] (2016)$^{2}_{S/R/G}$	\	\	\	\	0.690	0.822
MV-LSTM[88] (2016)$^{2,6}_{S/R/G}$	0.708	0.782	\	\	0.766	0.869
MatchPyramid[18] (2016)$^{2}_{S/I/G}$	\	\	\	\	0.764	0.867
aNMM[61] (2016)$^{3}_{A/I/G}$	0.750	0.811	\	\	\	\
Match-SRNN[69] (2016)$^{2}_{S/I/G}$	\	\	\	\	0.790	0.882
IARNN[86] (2016)$^{4}_{A/H/G}$	\	\	0.734	0.742	\	\
HD-LSTM[127] (2017)$^{6}_{S/R/G}$	0.750	0.815	\	\	\	\
CompAgg[87] (2017)$^{7}_{A/I/G}$	\	\	0.743	0.755	\	\
HyperQA[128] (2018)$^{8}_{S/R/G}$	0.770	0.825	0.712	0.727	\	\
MIX[95] (2018)$^{10}_{S/I/M}$	\	\	0.713	\	\	\

matrices in the attention channels to encapsulate rich structural patterns. aNMM adopts attention mechanism to encode question term importance for aggregating interaction matching features.

7. Trending Topics

In this section, we discuss several trending topics related to neural ranking models. Some of these topics are important but have not been well addressed in this field, while some are very promising directions for future research.

7.1. Indexing: from Re-ranking to Ranking

Modern search engines take advantage of a multi-stage cascaded architecture in order to efficiently provide accurate result lists to users. In more detail, there can be a stack of rankers, starting from an efficient high-recall model. Learning to rank models are often employed to model the last stage ranker whose goal is to re-rank a small set of documents retrieved by the early stage rankers. The main objective of these learning to rank models is to provide high-precision results.

Such a multi-stage cascaded architecture suffers from an error propagation problem. In other words, the errors initiated by the early stage rankers are propagated to the last stage. This clearly shows that multi-stage systems are not optimal. However, for efficiency reasons, learning to rank models cannot be used as the sole ranker to retrieve from large collections, which is a disadvantage for such models.

To address this issue, Zamani et al. [28] recently argued that the sparse nature of natural languages enables efficient term-matching retrieval models to take advantage of an *inverted index* data structure for efficient retrieval. Therefore, they proposed a standalone neural ranking model (SNRM) that learns high-dimensional sparse representations for queries and documents. In more detail, this type of model should optimize two objectives: (i) a relevance objective that maximizes the effectiveness of the model in terms of the retrieval

performance, and (*ii*) a sparsity objective that is equivalent to minimizing L_0 of the query and document representations. SNRM has shown superior performance compared to competitive baselines and has performed as efficiently as term-matching models, such as TF-IDF and BM25.

Learning inverted indexes has been also started to be explored in the database community. Kraska et al. [130] recently proposed to look at indexes as models. For example, a B-Tree-Index can be seen as a function that maps each key to a position of record in a sorted list. They proposed to replace traditional indexes used in databases with the indexes learned using deep learning technologies. Their models demonstrate a significant conflict reduction and memory footprint improvement.

Graph-based hashing and indexing algorithms have also attracted a considerable attention, which could be leveraged to index neural representations for the initial retrieval. For instance, Boytsov et al. [131] proposed to replace term-matching retrieval models with approximate nearest neighbor algorithms. Van Gysel et al. [132] used a similar idea to design an unsupervised neural retrieval model, however, their model architecture is not scalable to large document collections.

Moving from re-ranking a small set of documents to retrieving documents from a large collection is a recent research direction with a number of unanswered questions that require further investigation. For example, understanding and interpreting the learned neural representations has yet to be addressed. Furthermore, there is a known trade-off between efficiency and effectiveness in information retrieval systems, however, understanding this trade-off in learning inverted indexes requires further research. In addition, although index compression is a common technique in the search engine industry to reduce the size of the posting lists and improve efficiency, compression of the learned latent indexes is an unexplored area of research.

In summary, learning to index and developing effective and at the same time efficient retrieval models is a promising direction in neural IR research, however, we still face several open questions in this area.

7.2. Learning with External Knowledge

Most existing neural ranking models focus on learning the matching patterns between the two input texts. In recent years, some researchers have gone beyond matching textual objects by leveraging external knowledge to enhance the ranking performance. These research works can be grouped into two categories: 1) learning with external structured knowledge such as knowledge bases [133, 29, 134, 135, 136, 137]; 2) learning with external unstructured knowledge such as retrieved top results, topics or tags [30, 138, 139]. We now briefly review this work.

The first category of research explored improving neural ranking models with semantic information from knowledge bases. Liu et al. [133] proposed EDRM that incorporates entities in interaction-focused neural ranking models. EDRM first learns the distributed representations of entities using their semantics from knowledge bases in descriptions and types. Then the model matches documents to queries with both bag-of-words and bag-of-entities. Similar approaches were proposed by Xiong et al. [29], which also models queries and documents with word-based representations and entity-based representations. Nguyen et al. [134] proposed combining distributional semantics learned through neural networks and symbolic semantics held by extracted concepts or entities from text knowledge bases to enhance the learning algorithm of latent representations of queries and documents. Shen et al. [135] proposed the KABLSTM model, which leverages external knowledge from knowledge graphs to enrich the representational learning of QA sentences. Xu et al. [137] designed a Recall gate, where domain knowledge can be transformed into the extra global memory of LSTM, with the aim of enhancing LSTM by cooperating with its local memory to capture the implicit semantic relevance between sentences within conversations.

Beyond structured knowledge in knowledge bases, other research has explored how to integrate external knowledge from unstructured texts, which are more common for information on the Web. Yang et al. [30] studied response ranking in information-seeking conversations and proposed two effective meth-

ods to incorporate external knowledge into neural ranking models with pseudo-relevance feedback (PRF) and QA correspondence knowledge distillation. They proposed to extract the "correspondence" regularities between question and answer terms from retrieved external QA pairs as external knowledge to help response selection. Another representative work on integrating unstructured knowledge into neural ranking models is the KEHNN model proposed by Wu et al. [139], which defined prior knowledge as topics, tags, and entities related to the text pair. KEHNN represents global context obtained from external textual collection, and then exploits a knowledge gate to fuse the semantic information carried by the prior knowledge into the representation of words. Finally, it generates a knowledge enhanced representation for each word to construct the interaction matrix between text pairs.

In summary, learning with external knowledge is an active research area related to neural ranking models. More research efforts are needed to improve the effectiveness of neural ranking models with distilled external knowledge and to understand the role of external knowledge in ranking tasks.

7.3. Learning with Visualized Technology

We have discussed many neural ranking models in this survey under the textual IR scenario. There have also been a few studies showing that the textual IR problem could be solved visually. The key idea is that we can construct the matching between two inputs as an image so that we can leverage deep neural models to estimate the relevance based on visual features. The advantage of the matching image, compared with traditional matching matrix, is that it can keep the layout information of the original inputs so that many useful features such as spatial proximity, font size and colors could be modeled for relevance estimation. This is especially useful when we consider ad-hoc retrieval tasks on the Web where pages are often well designed documents with rich layout information.

Specifically, Fan et al. [31] proposed a visual perception model (ViP) to perceive visual features for relevance estimation. They first rendered the Web pages

into query-independent snapshots and query-dependent snapshots. Then, the visual features are learned through a combination of CNN and LSTM, inspired by users' reading behaviour. The results have demonstrated the effectiveness of learning the visual features of document for ranking problems. Zhang et al. [140] proposed a joint relevance estimation model which learns visual patterns, textual semantics and presentation structures jointly from screenshots, titles, snippets and HTML source codes of search results. Their results have demonstrated the viability of the visual features in search result page relevance estimation. Recently, Akker et al. [141] built a dataset for the LTR task with visual features, named Visual learning TO Rank (ViTOR). The ViTOR dataset consists of visual snapshots, non-visual features and relevance judgments for ClueWeb12 webpages and TREC Web Track queries. Their results have demonstrated that visual features can significantly improve the LTR performance.

In summary, solving the textual ranking problem through visualized technology is a novel and interesting direction. In some sense, this approach simulates human behavior as we also judge relevance through visual perception. The existing work has only demonstrated the effectiveness of visual features in some relevance assessment tasks. However, more research is needed to understand what can be learned by such visualized technology beyond those text-based methods, and what IR applications could benefit from such models.

7.4. Learning with Context

Search queries are often short and cannot precisely express the underlying information needs. To address this issue, a common strategy is to exploit *query context* to improve the retrieval performance. Different types of query context have been explored in the literature:

- Short-term history: the user past interactions with the system in the current search session [142, 143, 144].
- Long-term history: the historical information of the user's queries that is often used for web search personalization [145, 146].

- Situational context: the properties of the current search request, independent from the query content, such as location and time [147, 148].
- (Pseudo-) relevance feedback: explicit, implicit, or pseudo relevance signals for a given query can be used as the query context to improve the retrieval performance.

Although query context has been widely explored in the literature, incorporating query context into neural ranking models is relatively less studied. Zamani et al. [148] proposed a deep and wide network architecture in which the deep part of the model learns abstract representations for contextual features, while the wide part of the model uses raw contextual features in binary format in order to avoid information loss as a result of high-level abstraction. Ahmad et al. [149] incorporated short-term history information into a neural ranking model by multi-task training of document ranking and query suggestion. Short- and long-term history have been also used by Chen et al. [150] for query suggestion.

In addition, learning high-dimensional representation for pseudo-relevance feedback has been also studied in the literature. In this area, embedding-based relevance models [151] extend the original relevance models [121] by considering word embedding vectors. The word embedding vectors can be obtained from self-supervised algorithms, such as word2vec [152], or weakly supervised algorithms, such as relevance-based word embedding [111]. Zamani et al. [153] proposed RFMF, the first pseudo-relevance feedback model that learns latent factors from the top retrieved document. RFMF uses non-negative matrix factorization for learning latent representations for words, queries, and documents. Later on, Li et al. [119] extended existing neural ranking models, e.g., DRMM [21] and KNRM [85], by a neural pseudo-relevance feedback approach, called NPRF. The authors showed that in many cases extending a neural ranking model with NPRF leads to significant improvements. Zamani et al. [28] also made a similar conclusion by extending SNRM with pseudo-relevance feedback.

In summary, with the emergence of interactive or conversational search sys-

tem, context-aware ranking would be an indispensable technology in these scenarios. These exist several open research questions on how to incorporate query context information in neural ranking models. More research work is expected in this direction in the short future.

7.5. Neural Ranking Model Understanding

Deep learning techniques have been widely criticized as a "black box" which produces good results but no problem insights and explanations. Thus, how to understand and explain neural models has been an important topic in both Machine Learning and IR communities. To the best of our knowledge, the explainability of neural ranking models has not been fully studied. Instead, there have been a few papers on analyzing and understanding the empirical effect of different model components in IR tasks.

For example, Pang et al. [154] conducted an extensive analysis on the MatchPyramid model in ad-hoc retrieval and compared different kernals, pooling sizes, and similarity functions in terms of retrieval performance. Cohen et al. [155] extracted the internal representations of neural ranking models and evaluated their effectiveness in four natural language processing tasks. They find that topical relevance information is usually captured in the high-level layers of a neural model. Nie et al. [156] conducted empirical studies on the interaction-based neural ranking model to understand what have been learned in each neural network layer. They also notice that low-level network layers tend to capture detailed text information while high-level layers tend to have higher topical information abstraction.

While the paradigms of analyzing neural ranking models often rely on a deep understanding of specific model structure, Cohen et al. [22] argue that there are some general patterns of which types of neural models are more suitable for each IR task. For example, retrieval tasks with fine granularity (e.g., factoid QA) usually need higher levels of information abstraction and semantic matching, while retrieval tasks with coarse granularity (e.g., document retrieval) often rely on the exact matching or interaction between query words and document

words.

Overall, the research area on the explainability of neural ranking models is largely unexplored up till now. Some skepticism about neural ranking models is also related to this, e.g., what new things can be learned by neural ranking models? It is a very challenging and promising direction for researchers in neural IR.

8. Conclusion

The purpose of this survey is to summarize the current research status on neural ranking models, analyze the existing methodologies, and gain some insights for future development. We introduced a unified formulation over the neural ranking models, and reviewed existing models based on this formulation from different dimensions under model architecture and model learning. For model architecture analysis, we reviewed existing models to understand their underlying assumptions and major design principles, including how to treat the inputs, how to consider the relevance features, and how to make evaluation. For model learning analysis, we reviewed popular learning objectives and training strategies adopted for neural ranking models. To better understand the current status of neural ranking models on major applications, we surveyed published empirical results on the ad-hoc retrieval and QA tasks to conduct a comprehensive comparison. In addition, we discussed several trending topics that are important or might be promising in the future.

Just as there has been an explosion in the development of many deep learning based methods, research on neural ranking models has increased rapidly and broadened in terms of applications. We hope this survey can help researchers who are interested in this direction, and will motivate new ideas by looking at past successes and failures. Neural ranking models are part of the broader research field of neural IR, which is a joint domain of deep learning and IR technologies with many opportunities for new research and applications. We are expecting that, through the efforts of the community, significant breakthroughs

will be achieved in this domain in the near future, similar to those happened in computer vision or NLP.

9. Acknowledgments

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61425016 and 61722211, and the Youth Innovation Promotion Association CAS under Grants No. 20144310. This work was supported in part by the UMass Amherst Center for Intelligent Information Retrieval and in part by NSF IIS-1715095. 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] G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.

[2] S. E. Robertson, K. S. Jones, Relevance weighting of search terms, Journal of the American Society for Information science 27 (3) (1976) 129–146.

[3] T.-Y. Liu, Learning to rank for information retrieval, Found. Trends Inf. Retr. 3 (3) (2009) 225–331.

[4] H. Li, Learning to Rank for Information Retrieval and Natural Language Processing, Morgan & Claypool Publishers, 2011.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, T. Sainath, Deep neural networks for acoustic modeling in speech recognition, IEEE Signal Processing Magazine 29 (2012) 82–97.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.

[7] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436 EP –.

[8] Y. Goldberg, Neural network methods for natural language processing, Synthesis Lectures on Human Language Technologies 10 (1) (2017) 1–309.

[9] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473.

[10] N. Craswell, W. B. Croft, J. Guo, B. Mitra, M. de Rijke, Report on the sigir 2016 workshop on neural information retrieval (neu-ir), SIGIR Forum 50 (2) (2017) 96–103.

[11] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, J. Li, Deep learning for content-based image retrieval: A comprehensive study, in: Proceedings of the 22Nd ACM International Conference on Multimedia, MM '14, ACM, New York, NY, USA, 2014, pp. 157–166.

[12] E. Brenner, J. Zhao, A. Kutiyanawala, Z. Yan, End-to-end neural ranking for ecommerce product search: an application of task models and textual embeddings, arXiv preprint arXiv:1806.07296.

[13] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck, Learning deep structured semantic models for web search using clickthrough data, in: Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM '13, ACM, New York, NY, USA, 2013, pp. 2333–2338.

[14] Z. Lu, H. Li, A deep architecture for matching short texts, in: Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 1367–1375.

[15] R. Salakhutdinov, G. Hinton, Semantic hashing, International Journal of Approximate Reasoning 50 (7) (2009) 969–978.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 3111–3119.

[17] B. Hu, Z. Lu, H. Li, Q. Chen, Convolutional neural network architectures for matching natural language sentences, in: Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 2042–2050.

[18] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, X. Cheng, Text matching as image recognition, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[19] A. Severyn, A. Moschitti, Learning to rank short text pairs with convolutional deep neural networks, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, ACM, New York, NY, USA, 2015, pp. 373–382.

[20] K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. Mcnamara, A. Angert, E. Banner, V. Khetan, T. Mcdonnell, A. T. Nguyen, D. Xu, B. C. Wallace, M. Rijke, M. Lease, Neural information retrieval: At the end of the early years, Inf. Retr. 21 (2-3) (2018) 111–182.

[21] J. Guo, Y. Fan, Q. Ai, W. B. Croft, A deep relevance matching model for ad-hoc retrieval, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, ACM, New York, NY, USA, 2016, pp. 55–64.

[22] D. Cohen, Q. Ai, W. B. Croft, Adaptability of neural networks on varying granularity ir tasks, arXiv preprint arXiv:1606.07565.

[23] B. Mitra, F. Diaz, N. Craswell, Learning to match using local and distributed representations of text for web search, in: Proceedings of the

26th International Conference on World Wide Web, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2017, pp. 1291–1299.

[24] K. Hui, A. Yates, K. Berberich, G. de Melo, Pacrr: A position-aware neural ir model for relevance matching, arXiv preprint arXiv:1704.03940.

[25] X. Qiu, X. Huang, Convolutional neural tensor network architecture for community-based question answering, in: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, AAAI Press, 2015, pp. 1305–1311.

[26] R. Yan, Y. Song, H. Wu, Learning to respond with deep neural networks for retrieval-based human-computer conversation system, in: SIGIR, 2016.

[27] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, W. B. Croft, Neural ranking models with weak supervision, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, 2017, pp. 65–74.

[28] H. Zamani, M. Dehghani, W. B. Croft, E. Learned-Miller, J. Kamps, From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, ACM, New York, NY, USA, 2018, pp. 497–506.

[29] C. Xiong, J. Callan, T.-Y. Liu, Word-entity duet representations for document ranking, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, 2017, pp. 763–772.

[30] L. Yang, M. Qiu, C. Qu, J. Guo, Y. Zhang, W. B. Croft, J. Huang, H. Chen, Response ranking with deep matching networks and external

knowledge in information-seeking conversation systems, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, 2018, pp. 245–254.

[31] Y. Fan, J. Guo, Y. Lan, J. Xu, L. Pang, X. Cheng, Learning visual features from snapshots for web search, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 247–256.

[32] Z. Tang, G. H. Yang, Deeptilebars: Visualizing term distribution for neural information retrieval, arXiv preprint arXiv:1811.00606.

[33] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, X. Cheng, Deeprank: A new deep architecture for relevance ranking in information retrieval, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, ACM, New York, NY, USA, 2017, pp. 257–266.

[34] Y. Fan, J. Guo, Y. Lan, J. Xu, C. Zhai, X. Cheng, Modeling diverse relevance patterns in ad-hoc retrieval, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 375–384.

[35] N. Craswell, W. B. Croft, M. de Rijke, J. Guo, B. Mitra, Sigir 2017 workshop on neural information retrieval (neu-ir'17), in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, 2017, pp. 1431–1432.

[36] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, MS MARCO: A human generated machine reading comprehension dataset, CoRR abs/1611.09268. arXiv:1611.09268.

[37] Y. Yang, S. W.-t. Yih, C. Meek, Wikiqa: A challenge dataset for open-domain question answering, proceedings of the 2015 conference on empir-

ical methods in natural language processing Edition, ACL - Association for Computational Linguistics, 2015.

[38] L. Dietz, M. Verma, F. Radlinski, N. Craswell, TREC complex answer retrieval overview, in: Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017, 2017.

[39] Y. Fan, L. Pang, J. Hou, J. Guo, Y. Lan, X. Cheng, Matchzoo: A toolkit for deep text matching, arXiv preprint arXiv:1707.07270.

[40] X. He, J. Gao, L. Deng, Deep learning for natural language processing: Theory and practice (tutorial), 2014.

[41] B. Mitra, N. Craswell, Neural models for information retrieval, arXiv preprint arXiv:1705.01509.

[42] M. Grbovic, N. Djuric, V. Radosavljevic, F. Silvestri, N. Bhamidipati, Context- and content-aware embeddings for query rewriting in sponsored search, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, ACM, New York, NY, USA, 2015, pp. 383–392.

[43] R. Baeza-Yates, B. d. A. N. Ribeiro, et al., Modern information retrieval, New York: ACM Press; Harlow, England: Addison-Wesley,, 2011.

[44] G. W. Furnas, T. K. Landauer, L. M. Gomez, S. T. Dumais, The vocabulary problem in human-system communication, Commun. ACM 30 (11) (1987) 964–971.

[45] L. Zhao, J. Callan, Term necessity prediction, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, ACM, New York, NY, USA, 2010, pp. 259–268.

[46] S. Robertson, S. Walker, Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval, in: SIGIR '94, 1994.

[47] Y. Shen, X. He, J. Gao, L. Deng, G. Mesnil, A latent semantic model with convolutional-pooling structure for information retrieval, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, ACM, New York, NY, USA, 2014, pp. 101–110.

[48] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, R. Ward, Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval, IEEE/ACM Trans. Audio, Speech and Lang. Proc. 24 (4) (2016) 694–707.

[49] Y. Zheng, Z. Fan, Y. Liu, C. Luo, M. Zhang, S. Ma, Sogou-qcl: A new dataset with click relevance label, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 1117–1120.

[50] D. Mollá, J. L. Vicedo, Question answering in restricted domains: An overview, Computational Linguistics 33 (1) (2007) 41–61.

[51] A. Moschitti, L. Márquez, P. Nakov, E. Agichtein, C. Clarke, I. Szpektor, Sigir 2016 workshop webqa ii: Web question answering beyond factoids, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '16, ACM, New York, NY, USA, 2016, pp. 1251–1252.

[52] M. Richardson, Mctest: A challenge dataset for the open-domain machine comprehension of text, proceedings of the 2013 conference on empirical methods in natural language processing (emnlp 2013) Edition, 2013.

[53] E. M. Voorhees, D. M. Tice, Building a question answering test collection, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00, ACM, New York, NY, USA, 2000, pp. 200–207.

[54] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100, 000+ questions for machine comprehension of text, CoRR abs/1606.05250. arXiv:1606.05250.

[55] B. Mitra, G. Simon, J. Gao, N. Craswell, L. Deng, A proposal for evaluating answer distillation from web data.

[56] D. Cohen, L. Yang, W. B. Croft, Wikipassageqa: A benchmark collection for research on non-factoid answer passage retrieval, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, 2018, pp. 1165–1168.

[57] M. Keikha, J. H. Park, W. B. Croft, Evaluating answer passages using summarization measures, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14, ACM, New York, NY, USA, 2014, pp. 963–966.

[58] L. Yang, Q. Ai, D. Spina, R. Chen, L. Pang, W. B. Croft, J. Guo, F. Scholer, Beyond factoid QA: effective methods for non-factoid answer sentence retrieval, in: Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings, 2016, pp. 115–128.

[59] M. Feng, B. Xiang, M. R. Glass, L. Wang, B. Zhou, Applying deep learning to answer selection: A study and an open task, CoRR abs/1508.01585. arXiv:1508.01585.

[60] D. Wang, E. Nyberg, A long short-term memory model for answer sentence selection in question answering, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, 2015, pp. 707–712.

[61] L. Yang, Q. Ai, J. Guo, W. B. Croft, anmm: Ranking short answer texts with attention-based neural matching model, in: Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016, 2016, pp. 287–296.

[62] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, Z. Chen, Cqarank: Jointly model topics and expertise in community question answering, in: Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM '13, ACM, New York, NY, USA, 2013, pp. 99–108.

[63] A. Shtok, G. Dror, Y. Maarek, I. Szpektor, Learning from the past: Answering new questions with past answers, in: Proceedings of the 21st International Conference on World Wide Web, WWW '12, ACM, New York, NY, USA, 2012, pp. 759–768.

[64] P. Nakov, D. Hoogeveen, L. Màrquez, A. Moschitti, H. Mubarak, T. Baldwin, K. Verspoor, SemEval-2017 task 3: Community question answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval '17, Association for Computational Linguistics, Vancouver, Canada, 2017.

[65] D. Hoogeveen, K. M. Verspoor, T. Baldwin, Cqadupstack: A benchmark data set for community question-answering research, in: Proceedings of the 20th Australasian Document Computing Symposium, ACM, 2015, p. 3.

[66] A. Abujabal, R. S. Roy, M. Yahya, G. Weikum, Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters, arXiv preprint arXiv:1809.09528.

[67] X. Liu, C. Wang, Y. Leng, C. Zhai, Linkso: a dataset for learning to retrieve similar question answer pairs on software development forums,

in: Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering, ACM, 2018, pp. 2–5.

[68] Z. Wang, W. Hamza, R. Florian, Bilateral multi-perspective matching for natural language sentences, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, AAAI Press, 2017, pp. 4144–4150.

[69] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, X. Cheng, Match-srnn: Modeling the recursive matching structure with spatial rnn, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, AAAI Press, 2016, pp. 2922–2928.

[70] L. Chen, Y. Lan, L. Pang, J. Guo, J. Xu, X. Cheng, Ri-match: Integrating both representations and interactions for deep semantic matching, in: Information Retrieval Technology, Springer International Publishing, Cham, 2018, pp. 90–102.

[71] J. Gao, M. Galley, L. Li, Neural approaches to conversational AI, CoRR abs/1809.08267. arXiv:1809.08267.

[72] Z. Ji, Z. Lu, H. Li, An information retrieval approach to short text conversation, CoRR abs/1408.6988.

[73] A. Ritter, C. Cherry, W. B. Dolan, Data-driven response generation in social media, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 583–593.

[74] H. Wang, Z. Lu, H. Li, E. Chen, A dataset for research on short-text conversations, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 935–945.

[75] Y. Wu, W. Wu, C. Xing, M. Zhou, Z. Li, Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots, in: ACL '17, 2017.

[76] R. Lowe, N. Pow, I. Serban, J. Pineau, The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems, CoRR abs/1506.08909.

[77] X. Zhou, D. Dong, H. Wu, S. Zhao, D. Yu, H. Tian, X. Liu, R. Yan, Multi-view response selection for human-computer conversation, in: EMNLP, 2016.

[78] L. Yang, H. Zamani, Y. Zhang, J. Guo, W. B. Croft, Neural matching models for question retrieval and next question prediction in conversation, CoRR.

[79] R. Yan, Y. Song, X. Zhou, H. Wu, "shall I be your chat companion?": Towards an online human-computer conversation system, in: CIKM '16, 2016.

[80] R. Yan, D. Zhao, W. E., Joint learning of response ranking and next utterance suggestion in human-computer conversation system, in: SIGIR '17, 2017.

[81] C. Qu, L. Yang, W. B. Croft, J. Trippas, Y. Zhang, M. Qiu, Analyzing and characterizing user intent in information-seeking conversations., in: SIGIR '18, 2018.

[82] C. Qu, L. Yang, W. B. Croft, Y. Zhang, J. Trippas, M. Qiu, User intent prediction in information-seeking conversations, in: CHIIR '19, 2019.

[83] L. Shang, T. Sakai, Overview of the ntcir-12 short text conversation task.

[84] Z. Dai, C. Xiong, J. Callan, Z. Liu, Convolutional neural networks for soft-matching n-grams in ad-hoc search, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, ACM, New York, NY, USA, 2018, pp. 126–134.

[85] C. Xiong, Z. Dai, J. Callan, Z. Liu, R. Power, End-to-end neural ad-hoc ranking with kernel pooling, in: Proceedings of the 40th International

ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, 2017, pp. 55–64.

[86] B. Wang, K. Liu, J. Zhao, Inner attention based recurrent neural networks for answer selection, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2016, pp. 1288–1297.

[87] S. Wang, J. Jiang, A compare-aggregate model for matching text sequences, in: Proceedings of the 5th International Conference on Learning Representations, ICLR'17, 2017.

[88] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, X. Cheng, A deep architecture for semantic matching with multiple positional sentence representations, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, AAAI Press, 2016, pp. 2835–2841.

[89] W. Yang, H. Zhang, J. Lin, Simple applications of bert for ad hoc document retrieval, arXiv preprint arXiv:1903.10972.

[90] W. Yin, H. Schütze, Multigrancnn: An architecture for general matching of text chunks on multiple levels of granularity, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, 2015, pp. 63–73.

[91] Y. Nie, A. Sordoni, J.-Y. Nie, Multi-level abstraction convolutional model with weak supervision for information retrieval, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 985–988.

[92] J. Rao, W. Yang, Y. Zhang, F. Ture, J. J. Lin, Multi-perspective relevance matching with hierarchical convnets for social media search, national conference on artificial intelligence.

[93] Y. Nie, Y. Li, J. Nie, Empirical study of multi-level convolution models for ir based on representations and interactions (2018) 59–66.

[94] J. Huang, S. Yao, C. Lyu, D. Ji, Multi-granularity neural sentence model for measuring short text similarity, in: Database Systems for Advanced Applications, Springer International Publishing, Cham, 2017, pp. 439–455.

[95] H. Chen, F. X. Han, D. Niu, D. Liu, K. Lai, C. Wu, Y. Xu, Mix: Multi-channel information crossing for text matching, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, ACM, New York, NY, USA, 2018, pp. 110–119.

[96] W. Chen, T.-Y. Liu, Y. Lan, Z.-M. Ma, H. Li, Ranking measures and loss functions in learning to rank, in: Advances in Neural Information Processing Systems, 2009, pp. 315–323.

[97] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. N. Hullender, Learning to rank using gradient descent, in: Proceedings of the 22nd International Conference on Machine learning (ICML-05), 2005, pp. 89–96.

[98] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, H. Li, Listwise approach to learning to rank: theory and algorithm, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 1192–1199.

[99] M. Taylor, J. Guiver, S. Robertson, T. Minka, Softrank: optimizing non-smooth rank metrics, in: Proceedings of WSDM'08, ACM, 2008, pp. 77–86.

[100] C. J. Burges, From ranknet to lambdarank to lambdamart: An overview, Learning 11 (2010) 23–581.

[101] Q. Ai, K. Bi, J. Guo, W. B. Croft, Learning a deep listwise context model for ranking refinement, in: The 41st International ACM SIGIR Conference

on Research & Development in Information Retrieval, ACM, 2018, pp. 135–144.

[102] Q. Ai, X. Wang, N. Golbandi, M. Bendersky, M. Najork, Learning groupwise scoring functions using deep neural networks, arXiv preprint arXiv:1811.04415.

[103] Q. Ai, J. Mao, Y. Liu, W. B. Croft, Unbiased learning to rank: Theory and practice, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 2305–2306.

[104] X. Liu, J. Gao, X. He, L. Deng, K. Duh, Y.-Y. Wang, Representation learning using multi-task deep neural networks for semantic classification and information retrieval.

[105] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, B. Tseng, Multi-task learning for boosting with application to web search ranking, in: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 1189–1198.

[106] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[107] D. Cohen, B. Mitra, K. Hofmann, W. B. Croft, Cross domain regularization for neural ranking models using adversarial learning, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 1025–1028.

[108] H. Zamani, M. Dehghani, F. Diaz, H. Li, N. Craswell, Sigir 2018 workshop on learning from limited or noisy data for information retrieval, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 1439–1440.

[109] N. Asadi, D. Metzler, T. Elsayed, J. Lin, Pseudo test collections for learning web search ranking functions, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, ACM, New York, NY, USA, 2011, pp. 1073–1082.

[110] H. Zamani, W. B. Croft, J. S. Culpepper, Neural query performance prediction using weak supervision from multiple signals, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 105–114.

[111] H. Zamani, W. B. Croft, Relevance-based word embedding, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, 2017, pp. 505–514.

[112] D. Cohen, J. Foley, H. Zamani, J. Allan, W. B. Croft, Universal approximation functions for fast learning to rank: Replacing expensive regression forests with simple feed-forward networks, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 1017–1020.

[113] M. Szummer, E. Yilmaz, Semi-supervised learning to rank with preference regularization, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11, ACM, New York, NY, USA, 2011, pp. 269–278.

[114] K. Duh, K. Kirchhoff, Learning to rank with partially-labeled data, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, ACM, New York, NY, USA, 2008, pp. 251–258.

[115] X. Zhang, B. He, T. Luo, Training query filtering for semi-supervised learning to rank with pseudo labels, World Wide Web 19 (5) (2016) 833–864.

[116] M. Dehghani, A. Severyn, S. Rothe, J. Kamps, Avoiding your teacher's mistakes: Training neural networks with controlled weak supervision, CoRR abs/1711.00313. arXiv:1711.00313.

[117] B. Li, P. Cheng, L. Jia, Joint learning from labeled and unlabeled data for information retrieval, in: Proceedings of the 27th International Conference on Computational Linguistics, COLING '18, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 293–302.

[118] K. Hui, A. Yates, K. Berberich, G. de Melo, Co-pacrr: A context-aware neural ir model for ad-hoc retrieval, in: Proceedings of the eleventh ACM international conference on web search and data mining, ACM, 2018, pp. 279–287.

[119] C. Li, Y. Sun, B. He, L. Wang, K. Hui, A. Yates, L. Sun, J. Xu, NPRF: A neural pseudo relevance feedback framework for ad-hoc information retrieval, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP '18, 2018.

[120] J. M. Ponte, W. B. Croft, A language modeling approach to information retrieval, Ph.D. thesis, University of Massachusetts at Amherst (1998).

[121] V. Lavrenko, W. B. Croft, Relevance based language models, in: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2001, pp. 120–127.

[122] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 133–142.

[123] K. Hui, A. Yates, K. Berberich, G. de Melo, A position-aware deep model for relevance matching in information retrieval, arXiv preprint arXiv:1704.03940.

[124] M. Wang, N. A. Smith, T. Mitamura, What is the jeopardy model? a quasi-synchronous grammar for qa, in: Proceedings of the 2007 Joint

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007.

[125] L. Yu, K. M. Hermann, P. Blunsom, S. Pulman, Deep learning for answer sentence selection, CoRR abs/1412.1632. arXiv:1412.1632.

[126] J. Rao, H. He, J. Lin, Noise-contrastive estimation for answer selection with deep neural networks, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, 2016.

[127] Y. Tay, M. C. Phan, A. T. Luu, S. C. Hui, Learning to rank question answer pairs with holographic dual LSTM architecture, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017, 2017, pp. 695–704.

[128] Y. Tay, L. A. Tuan, S. C. Hui, Hyperbolic representation learning for fast and efficient neural question answering, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018, 2018, pp. 583–591.

[129] W. Yih, M. Chang, C. Meek, A. Pastusiak, Question answering using enhanced lexical semantic models, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria,, The Association for Computer Linguistics, 2013, pp. 1744–1753.

[130] T. Kraska, A. Beutel, E. H. Chi, J. Dean, N. Polyzotis, The case for learned index structures, in: Proceedings of the 2018 International Conference on Management of Data, SIGMOD '18, ACM, New York, NY, USA, 2018, pp. 489–504.

[131] L. Boytsov, D. Novak, Y. Malkov, E. Nyberg, Off the beaten path: Let's replace term-based retrieval with k-nn search, in: Proceedings of the 25th

ACM International on Conference on Information and Knowledge Management, CIKM '16, ACM, New York, NY, USA, 2016, pp. 1099–1108.

[132] C. V. Gysel, M. de Rijke, E. Kanoulas, Neural vector spaces for unsupervised information retrieval, ACM Trans. Inf. Syst. 36 (4) (2018) 38:1–38:25.

[133] Z. Liu, C. Xiong, M. Sun, Z. Liu, Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2018, pp. 2395–2405.

[134] G. Nguyen, L. Tamine, L. Soulier, N. Bricon-Souf, Toward a deep neural approach for knowledge-based IR, CoRR abs/1606.07211. arXiv:1606.07211.

[135] Y. Shen, Y. Deng, M. Yang, Y. Li, N. Du, W. Fan, K. Lei, Knowledge-aware attentive neural network for ranking question answer pairs, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 901–904.

[136] X. Song, F. Feng, X. Han, X. Yang, W. Liu, L. Nie, Neural compatibility modeling with attentive knowledge distillation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 5–14.

[137] Z. Xu, B. Liu, B. Wang, C. Sun, X. Wang, Incorporating loose-structured knowledge into LSTM with recall gate for conversation modeling, CoRR abs/1605.05110. arXiv:1605.05110.

[138] M. Ghazvininejad, C. Brockett, M. Chang, B. Dolan, J. Gao, W. Yih, M. Galley, A knowledge-grounded neural conversation model, in: Pro-

ceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), 2018, pp. 5110–5117.

[139] Y. Wu, W. Wu, C. Xu, Z. Li, Knowledge enhanced hybrid neural network for text matching, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), 2018, pp. 5586–5593.

[140] J. Zhang, Y. Liu, S. Ma, Q. Tian, Relevance estimation with multiple information sources on search engine result pages, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 627–636.

[141] B. v. d. Akker, I. Markov, M. de Rijke, Vitor: Learning to rank webpages based on visual features, arXiv preprint arXiv:1903.02939.

[142] X. Shen, B. Tan, C. Zhai, Context-sensitive information retrieval using implicit feedback, SIGIR '05, ACM, New York, NY, USA, 2005, pp. 43–50.

[143] Y. Ustinovskiy, P. Serdyukov, Personalization of web-search using short-term browsing context, CIKM '13, ACM, New York, NY, USA, 2013, pp. 1979–1988.

[144] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, H. Li, Context-aware ranking in web search, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, ACM, New York, NY, USA, 2010, pp. 451–458.

[145] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, X. Cui, Modeling the impact of short- and long-term behavior on search personalization, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, ACM, New York, NY, USA, 2012, pp. 185–194.

[146] N. Matthijs, F. Radlinski, Personalizing web search using long term browsing history, in: Proceedings of the Fourth ACM International Conference

on Web Search and Data Mining, WSDM '11, ACM, New York, NY, USA, 2011, pp. 25–34.

[147] P. N. Bennett, F. Radlinski, R. W. White, E. Yilmaz, Inferring and using location metadata to personalize web search, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, ACM, New York, NY, USA, 2011, pp. 135–154.

[148] H. Zamani, M. Bendersky, X. Wang, M. Zhang, Situational context for ranking in personal search, in: Proceedings of the 26th International Conference on World Wide Web, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2017, pp. 1531–1540.

[149] W. U. Ahmad, K.-W. Chang, H. Wang, Multi-task learning for document ranking and query suggestion, in: Proceedings of the Sixth International Conference on Learning Representations, ICLR '18, 2018.

[150] W. Chen, F. Cai, H. Chen, M. de Rijke, Attention-based hierarchical neural query suggestion, in: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 1093–1096.

[151] H. Zamani, W. B. Croft, Embedding-based query language models, in: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR '16, 2016, pp. 147–156.

[152] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

[153] H. Zamani, J. Dadashkarimi, A. Shakery, W. B. Croft, Pseudo-relevance feedback based on matrix factorization, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, 2016, pp. 1483–1492.

[154] L. Pang, Y. Lan, J. Guo, J. Xu, X. Cheng, A study of matchpyramid models on ad-hoc retrieval, arXiv preprint arXiv:1606.04648.

[155] D. Cohen, B. O'Connor, W. B. Croft, Understanding the representational power of neural retrieval models using nlp tasks, in: Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ACM, 2018, pp. 67–74.

[156] Y. Nie, Y. Li, J.-Y. Nie, Empirical study of multi-level convolution models for ir based on representations and interactions, in: Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ACM, 2018, pp. 59–66.