DeepRank A New Deep Architecture for Relevance Ranking in Information Retrieval

Liang Pang[†*], Yanyan Lan[†*], Jiafeng Guo[†*], Jun Xu[†*], Jingfang Xu[‡], Xueqi Cheng[†*]
pl8787@gmail.com,{lanyanyan,guojiafeng,junxu,cxq}@ict.ac.cn,xujingfang@sogou-inc.com
†CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
∗University of Chinese Academy of Sciences, Beijing, China
‡Sogou Inc, Beijing, China

ABSTRACT

This paper concerns a deep learning approach to relevance ranking in information retrieval (IR). Existing deep IR models such as DSSM and CDSSM directly apply neural networks to generate ranking scores, without explicit understandings of the relevance. According to the human judgement process, a relevance label is generated by the following three steps: 1) relevant locations are detected; 2) local relevances are determined; 3) local relevances are aggregated to output the relevance label. In this paper we propose a new deep learning architecture, namely DeepRank, to simulate the above human judgment process. Firstly, a detection strategy is designed to extract the relevant contexts. Then, a measure network is applied to determine the local relevances by utilizing a convolutional neural network (CNN) or two-dimensional gated recurrent units (2D-GRU). Finally, an aggregation network with sequential integration and term gating mechanism is used to produce a global relevance score. DeepRank well captures important IR characteristics, including exact/semantic matching signals, proximity heuristics, query term importance, and diverse relevance requirement. Experiments on both benchmark LETOR dataset and a large scale clickthrough data show that DeepRank can significantly outperform learning to ranking methods, and existing deep learning methods.

CCS CONCEPTS

•**Information systems** →**Retrieval models and ranking;**

KEYWORDS

Deep Learning; Ranking; Text Matching; Information Retrieval

1 INTRODUCTION

Relevance ranking is a core problem of information retrieval. Given a query and a set of candidate documents, a scoring function is usually utilized to determine the relevance degree of a document with respect to the query. Then a ranking list is produced by sorting in descending order of the relevance score. Modern learning to rank approach applies machine learning techniques to the ranking function, which combines different kinds of human knowledge (i.e. relevance features such as BM25 [27] and PageRank [22]) and therefore has achieved great improvements on the ranking performances [18]. However, a successful learning to rank algorithm usually relies on effective handcrafted features for the learning process. The feature engineering work is usually time-consuming, incomplete and over-specified, which largely hinder the further development of this approach [11].

Recently, deep learning approach [17] has shown great success in many machine learning applications such as speech recognition, computer vision, and natural language processing (NLP), owing to their ability of automatically learning the effective data representations (features). Therefore, a new direction of Neural IR is proposed to resort to deep learning for tackling the feature engineering problem of learning to rank, by directly using only automatically learned features from raw text of query and document. There have been some pioneer work, including DSSM [13], CDSSM [29], and DRMM [11]. Both DSSM and CDSSM directly apply deep neural networks to obtain the semantic representations of query and document, and the ranking score is produced by computing their cosine similarity. Guo et al. [11] argued that DSSM and CDSSM only consider the semantic matching between query and document, but ignore the more important relevance matching characteristics, such as exact matching signals, query term importance, and diverse matching requirement [27]. Thus they proposed another deep architecture, i.e. DRMM, to solve this problem. However, DRMM does not explicitly model the relevance generation process, and fails to capture important IR characteristics such as passage retrieval intrinsics [19] and proximity heuristics [31].

Inspired by the human judgement process, we propose a new deep learning architecture, namely DeepRank, to better capture the relevance intrinsics. According to the illustration in [33], the human judgement process can be divided into three steps. Human annotators first scan the whole document to detect the relevant locations. Then the local relevance for each detected location is decided. Finally, those local relevances are combined to form the global relevance of the entire document. Consequently, DeepRank contains three parts to simulate the human judgement process, by tackling the following three problems:

Where does the relevance occur? According to the query-centric assumption proposed in [33], the relevant information for a query

CIKM'17, November 6–10, 2017, Singapore.
 ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: http://dx.doi.org/10.1145/3132847.3132914

only locates in the contexts around query terms. Therefore, the context with a query term at the center position, namely query-centric context, is recognized as the relevant location in the detection step.

How to measure the local relevance? After the detection step, a measure network is utilized to determine the local relevance between query and each query-centric context. Firstly, a tensor is constructed to incorporate both the word representations of query/query-centric context, and the interactions between them. Then a CNN or 2D-GRU is applied on the tensor to output the representation of the local relevance. In this way, important IR characteristics such as exact/semantic matching signals, passage retrieval intrinsics, and proximity heuristics can be well captured.

How to aggregate such local relevances to determine the global relevance score? As shown by [33], two factors are important for user's complex principles of aggregating local relevances, i.e. query term importance [6] and diverse relevance requirement [27]. Therefore we propose to first aggregate local relevances at query term level, and then make the combination by considering weights of different terms, via a term gating network. To obtain the term level relevance, we first group the query-centric contexts with the same central word together. Then a recurrent neural network (RNN) such as GRU [4] or LSTM [10] is utilized to aggregate such local relevances sequentially, by further considering the position information of these query-centric contexts in the whole document.

Above all, DeepRank is a new architecture composed of three components, i.e. a detection strategy, a measure network with CNN/2D-GRU, and an aggregation network with term gating and RNN inside. Therefore, DeepRank can be trained end-to-end with the pairwise ranking loss via stochastic gradient descent. We conduct experiments on both benchmark LETOR4.0 data and a large scale clickthrough data collected from a commercial search engine. The experimental results show that: 1) Existing deep IR methods such as DSSM, CDSSM, and DRMM perform much worse, if using only automatically learned features, than the pairwise and listwise learning to rank methods. 2) DeepRank significantly outperforms not only all the existing deep IR models but also all the pairwise and listwise learning to rank baseline methods. 3) If we incorporate handcrafted features into the model, as did in SQA [28], DeepRank will be further improved, and the performance is better than SQA. We also conduct a detailed experimental analysis on DeepRank to investigate the influences of different settings.

To the best of our knowledge, DeepRank is the first deep IR model to outperform existing learning to rank models.

2 RELATED WORK

We first review related work on relevance ranking for IR, including learning to rank methods and deep learning methods.

2.1 Learning to Rank Methods

In the past few decades, machine learning techniques have been applied to IR, and gained great improvements to this area. This direction is called learning to rank. Major learning to rank methods can be grouped into three categories: pointwise, pairwise and listwise approach. Different approaches define different input and output spaces, use different hypotheses, and employ different loss functions. Pointwise approach, such as logistic regression [8], inputs a feature vector of each single document and outputs the relevance degree of each single document. Pairwise approach, such as RankSVM [14] and RankBoost [7], inputs pairs of documents, both represented by feature vectors and outputs the pairwise preference between each pair of documents. Listwise approach, such as ListNet [2], AdaRank [34] and LambdaMart [1], inputs a set of document features associated with query and outputs the ranked list. All these approaches focus on learning the optimal way of combining features through discriminative training. However, a successful learning to rank algorithm relies on effective handcrafted features for the learning process. The feature engineering work is usually time-consuming, incomplete and over-specified, which largely hinder the further development of this direction [11].

2.2 Deep Learning Methods

Recently, deep learning techniques have been applied to IR, for automatically learning effective ranking features. Examples include DSSM [13], CDSSM [29], and DRMM [11]. DSSM uses a deep neural network (DNN) to map both query and document to a common semantic space. Then the relevance score is calculated as the cosine similarity between these two vectors. Rather than using DNN, CDSSM is proposed to use CNN to better preserve the local word order information when capturing contextual information of query/document. Then max-pooling strategies are adopted to filter the salient semantic concepts to form a sentence level representation. However, DSSM and CDSSM view IR as a semantic matching problem, and focus on generating a good sentence level representations for query and document. They ignore the important intrinsics of relevance ranking in IR. Guo et al. [11] first point out the differences between semantic matching and relevance matching. They propose a new deep learning architecture DRMM to model IR's own characteristics, including exact matching signals, query terms importance, and diverse matching requirement. Specifically, DRMM first builds local interactions between each pair of words from query and document based on word embeddings, and then maps the local interactions to a matching histogram for each query term. Then DRMM employs DNN to learn hierarchical matching patterns. Finally, the relevance score is generated by aggregating the term level scores via a term gating network. Though DRMM has made the first step to design deep learning model specially for IR, it does not explicitly model the relevance generation process of human. It also fails to model the important IR intrinsics such as passage retrieval strategies and proximity heuristics.

Another related sort of deep models, flourishing in NLP, provide a new way of thinking if we treat IR task as a general text matching task, i.e. query matches document. These work can be mainly categorized as representation focused models and interaction focused models. The representation focused models try to build a good representation for each single text with a neural network, and then conduct matching between the two vectors. The DSSM and CDSSM mentioned above belong to this category. There are also some other ones such as ARC-I [12] model, which builds on word embeddings and makes use of convolutional layers and pooling layers to extract compositional text representation. The interaction focused models first build the local interactions between two texts, and