

A Deep Investigation of Deep IR Models

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Xueqi Cheng

CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,

Chinese Academy of Sciences

Beijing, China

pangliang@software.ict.ac.cn, {lanyanyan,guojafeng,junxu,cxq}@ict.ac.cn

ABSTRACT

The effective of information retrieval (IR) systems have become more important than ever. Deep IR models have gained increasing attention for its ability to automatically learning features from raw text; thus, many deep IR models have been proposed recently. However, the learning process of these deep IR models resemble a black box. Therefore, it is necessary to identify the difference between automatically learned features by deep IR models and hand-crafted features used in traditional learning to rank approaches. Furthermore, it is valuable to investigate the differences between these deep IR models. This paper aims to conduct a deep investigation on deep IR models. Specifically, we conduct an extensive empirical study on two different datasets, including Robust and LETOR4.0. We first compared the automatically learned features and hand-crafted features on the respects of query term coverage, document length, embeddings and robustness. It reveals a number of disadvantages compared with hand-crafted features. Therefore, we establish guidelines for improving existing deep IR models. Furthermore, we compare two different categories of deep IR models, i.e. representation-focused models and interaction-focused models. It is shown that two types of deep IR models focus on different categories of words, including topic-related words and query-related words.

CCS CONCEPTS

•Information systems →Retrieval models and ranking;

KEYWORDS

Deep Learning; Ranking; Text Matching; Information Retrieval

1 INTRODUCTION

Relevance ranking is the core problem in information retrieval (IR) system, which is to determine the relevance score for a document with respect to a particular query. Traditional approaches to tackle this problem include heuristic retrieval models and learning to rank approach. Heuristic retrieval models, such as TF-IDF [12] and BM25 [11], propose to incorporate human knowledge on relevance into the design of ranking function. Modern learning to rank approach currently turns to apply machine learning techniques to

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGIR 2017 Workshop on Neural Information Retrieval (Neu-IR'17), August 7–11, 2017, Shinjuku, Tokyo, Japan

© 2017 Copyright held by the owner/author(s). 123-4567-24-567/08/06.

DOI: 10.475/123.4

the ranking function, which combines different kinds of human knowledge (relevance features such as BM25 and PageRank) and therefore has achieved great improvements on the ranking performances [6]. However, a successful learning to rank algorithm usually relies on effective hand-crafted 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 [3].

Deep IR models have gained increasing attention for its ability to automatically learning features from raw text of query and document. Therefore, many deep IR models have been proposed to solve relevance ranking problem only considering the query and document textual data. As [3] has mentioned, deep IR models can be categorized into two branches, namely representation-focused models and interaction-focused models, depending on the different structures. However, deep IR models are the end-to-end system, the learning process of the deep IR models are still resemble a black box. Thus, it is curious to us that the different between automatically learned features by deep IR models and hand-crafted features used in traditional learning to rank approaches, and the differences between these deep IR models.

In this paper, we conduct a deep investigation on deep IR models under two aspects. Firstly, we compare the automatically learned features with hand-crafted features. Hand-crafted features, including TF-IDF, BM25 and other traditional retrieval models, are combined human knowledge and proven to follow some heuristic retrieval constrains [2]. Incorporate with these constrains, we check the properties of deep IR models, for example query term coverage, document length and embedding affect the performance of deep IR models. Additionally, We pick out bad cases from test dataset to conduct error analysis. For each group of bad cases, we establish guidelines for improving existing deep IR models. We also show that the robustness of automatically features are stronger than hand-crafted features. Secondly, we compare the differences between two kinds of deep IR models. By visualizing the pooling words of these deep IR models, it is shown that representation-focused models focus on topic-related words and interaction-focused models focus on query-related words. A synthetic experiment further prove these properties.

The rest of the paper is organized as follows. In Section 2, we introduce two types of existing deep IR models. In Section 3, we introduce the datasets for performance evaluation. Section 4 compares the differences between the automatically learned features and the hand-crafted features and Section 5 compares the differences between two types of deep IR models. Section 6 concludes the paper.

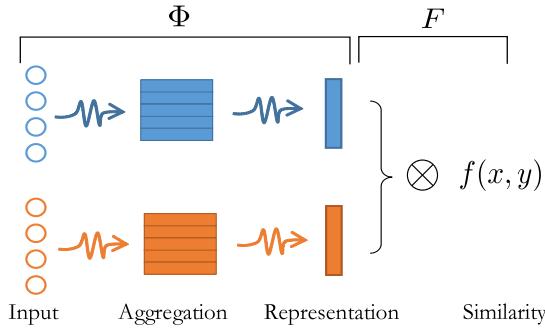


Figure 1: Representation-Focused Models

2 EXISTING DEEP IR MODEL

The core problem of information retrieval is to determine the relevance score for a document with respect to a particular query, which can be formalized as the follows as indicated in [4] and [8]. Given a query $Q = \{q_1, \dots, q_m\}$ and a document $D = \{w_1, \dots, w_n\}$, where q_i and w_j stand for the i -th and j -th words in the query and document respectively, the degree of relevance is usually measured as a score produced by a scoring function based on the representations of the query and document:

$$match(Q, D) = F(\Phi(Q), \Phi(D)), \quad (1)$$

where Φ is a function to map query/document to a vector, and F is a scoring function for modeling the interactions between them.

Deep IR models propose to automatically learn relevance features from raw text data, i.e. Q and D . Considering different structures, existing deep models can be categorized into two kinds: representation-focused models and interaction-focused models. The representation-focused models propose to focus on the learning parameters of function Φ , while interaction-focused models put more efforts on learning parameters of function F .

2.1 Representation-Focused Models

Representation-focused models try to build a good representation for query/document with a deep neural network, and then conduct matching between two abstract representation vector. In this approach, Φ is a relatively complex representation mapping function from text to vector, while F is a relatively simple matching function. For example, in DSSM [5], Φ is a feed forward neural network with letter trigram representation as the input, while F is the cosine similarity function. In CDSSM [13], Φ is a convolutional neural network (CNN) with letter trigram representation as the input, while F is the cosine similarity function. In ARC-I [4], Φ is a CNN with word embeddings as the input, while F is a multi-layer perceptron (MLP). Without loss of generality, all the model architectures of representation-focused models can be viewed as a Siamese (symmetric) architecture over the text inputs, as shown in Figure 1.

2.2 Interaction-Focused Models

Interaction-focused models first build the local interactions between query and document, based on basic representations, and then use

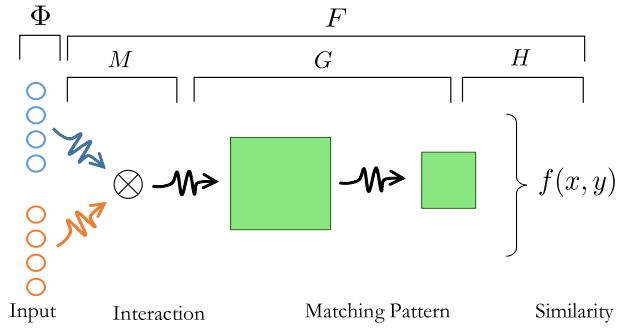


Figure 2: Interaction-Focused Models

deep neural networks to learn the complex interaction patterns for relevance. In this approach, Φ is usually a simple mapping function to map query and document to a sequence of words or word vectors, while F is a complex deep model with many learnable parameters. Typically, function F can be represented as the compound function of H , G and M , i.e. $F = H \circ G \circ M$, and the scoring function can be written as the following form:

$$match(Q, D) = H \circ G \circ M(\Phi(Q), \Phi(D)), \quad (2)$$

where M is a function used to obtain the local interactions between the representation of Q and D , H is a deep neural network to obtain the abstract interaction patterns, and G is an aggregation function to obtain the relevance score based on the interaction patterns.

For example, in DeepMatch [7], Φ is an identical function which maintains the representation of query/document as a sequences of words, M is a basic interaction function to output parallel texts, i.e. the set of interacting pairs of words from Q and D , G is a feed forward neural network constructed by a topic model over the parallel texts, and H is a logistic regression unit to summarize the decision to obtain the final relevance score. In ARC-II [4], Φ is a mapping function to map query/document to a sequence of word embeddings, M is the 1-D convolution operation over each patch of words from Q and D , G is a CNN, and H is an MLP. In MatchPyramid [8], Φ maps query/document to a sequence of word vectors, M is the similarity function between each word pair from Q and D to output a word-level interaction matrix, G is a CNN, and H is an MLP. In Match-SRNN, Φ is the same as that in MatchPyramid, M is a tensor operation to incorporate high dimensional word level interactions, G is a 2D-GRU, and H is an MLP. Without loss of generality, all the interaction focused models can be viewed as a deep architecture over the local interaction matrix, as shown in Figure 2.

3 DATASET

In this section, we introduce two datasets used for model analysis, namely Robust Dataset, LETOR 4.0 Dataset. The statistics of these datasets are shown in Table 1. We can see that these datasets represent different sizes and genres of heterogeneous text collections.

3.1 Robust Dataset

Robust data is a small news dataset. Its topics are collected from TREC Robust Track 2004. We made use of the title of each TREC

Table 1: Statistics on Robust and LETOR 4.0 .

Dataset	#Query	#Doc	#Relevance
Robust	250	12240	12881
LETOR 4.0	1501	57899	61480

topic in our experiments. The retrieval experiments on this dataset are implemented using the Galago Search Engine¹. During indexing and retrieval, both documents and query words are white-space tokenized, lower-cased, and stemmed using the Krovetz stemmer. Stopword removal is performed on query words during retrieval using the INQUERY stop list.

3.2 LETOR 4.0 Dataset

LETOR4.0 dataset [9] is a benchmark data for evaluating learning to rank methods. This dataset is sampled from the .GOV2 corpus using the TREC 2007 Million Query track queries. This dataset contains two subsets, i.e. MQ2007 and MQ2008. In this paper, we use MQ2007 for evaluation because it is much larger than MQ2008. MQ2007 contains 1692 queries and 65,323 documents, which is much larger than Robust. Each query and document pair in this dataset is represented as a vector using 46 different features. The separation of training, validation and testing set are set to default. The reason to choose LETOR4.0 beyond Robust lie in that: 1) the LETOR4.0 data is relatively large (especially for the query number), therefore it is more appropriate for training a deep learning model; 2) the features have already been extracted, therefore it is convenient to conduct comparisons with learning to rank baselines.

4 COMPARISONS WITH HAND-CRAFTED FEATURES

Learning to rank approaches with hand-crafted features have achieved a great success in information retrieval. In these hand-crafted features, BM25 and language model are the strong baselines in information retrieval. It is mainly because these traditional models satisfied several heuristic retrieval constraints proposed by Fang et al. [2]. These constraints reveal importance properties in information retrieval. They are 1) Term Frequency Constraints (TFC1 / TCF2); 2) Term Discrimination Constraint (TDC); 3) Length Normalization Constraints (LNC1 / LNC2); 4) TF-LENGTH Constraint (TF-LNC). Empirical results show that when a constraint is not satisfied, it often indicates non-optimality of the method.

In this section, our aim is to find out the differences between deep IR models and hand-crafted features. Firstly, bad cases are categorized to identify the weakness of deep IR models. Then we make use of heuristic retrieval constraints to explain the disadvantages of the deep IR models, such as query term coverage problem, document length problem and embedding semantic abuse problem. Lastly, we point out that the advantage of the deep IR models is the robustness of the automatically learnt features.

Table 2: Performance comparison of deep IR models and hand-crafted features on Robust, LETOR 4.0.

Robust					
Model	NDCG@1	NDCG@10	P@1	P@10	MAP
BM25-TITLE	0.563	0.445	0.563	0.402	0.255
LM.JM-TITLE	0.560	0.443	0.560	0.400	0.253
ARC-I	0.124	0.138	0.124	0.132	0.050
MATCHPYRAMID	0.364	0.242	0.364	0.240	0.164
LETOR 4.0					
Model	NDCG@1	NDCG@10	P@1	P@10	MAP
BM25-TITLE	0.358	0.414	0.427	0.366	0.450
LM.JM-TITLE	0.300	0.374	0.359	0.329	0.421
ARC-I	0.310	0.386	0.376	0.364	0.417
MATCHPYRAMID	0.362	0.409	0.428	0.371	0.434

4.1 Performance Comparison

Before we conduct error analysis, we first overview the performance of two kinds of deep IR models and two strong hand-crafted features, namely BM25 and language model, shown in Table 2.

The experimental result shows that deep IR models perform worse than the hand-crafted features, especially compared with BM25. Furthermore, we find that the performance gap between deep IR models and hand-crafted features is large in Robust dataset compared with LETOR 4.0 dataset. The main reason lays to the small size of the dataset, which go against the data-driven mechanism in deep learning. We reduce the performance gap by increasing the size of the dataset in LETOR 4.0, and we believe that the larger dataset will result in better performance, even better than the hand-crafted features.

Apart from the affection of dataset size, other heuristic problems are found by conducting error analysis. We will demonstrate these heuristic problem in the next section.

4.2 Error Analysis

4.2.1 Query term coverage problem. With analyzing number of bad cases of deep IR model results, we find that in the most of the cases, deep IR models are hard to satisfy the Term Discrimination Constraint (TDC) [2]. TDC ensures that given a fixed number of occurrences of query terms, we favor a document that has more occurrences of discriminative terms. However, for both Arc-I and MatchPyramid, we never intentionally designed a network structure to distinguish two same query terms matching from two discriminative query terms matching. Although, it is possible for these two models to learn from data, the limitation is the size of the dataset, which make it hard for deep IR models to learn TDC without any prior information.

We pick out one example, shown in Figure 3, to demonstrate this issue. Here the query is “tooth fairy museum”. In the left part of Figure 3, the document only contains query terms “museum” and “fairy”, where term “fairy” occurs more than 10 times. While in the right part of Figure 3, the document contains all three query terms, but the total number of query terms is only 6 times. With the constrain of TDC, a good IR model prefer to rank right side

¹<http://www.lemurproject.org/galago.php>

document higher than left side. However, the deep IR models make the opposite decision.

Suggestions: In order to recover query term coverage problem we propose two suggestions for representation-focused model and interaction-focused model respectively. 1) For representation-focused models, an attention mechanism turns to be a useful strategy to distinguish different query terms. 2) For interaction-focused models, pooling across each query term rows in the interaction matrix is helpful for considering different query terms individually.

4.2.2 Document length problem. Another type of errors relates to the general preprocessing of document length limitation before feed it into the deep IR models. With the limitation of memory and time, documents tailor to a maximum length. It always make sense in paraphrase identification tasks and question answering tasks, which have the similar text length. But in information retrieval tasks, documents have variance length, ranging from 10 to 10,000 words. Directly cut off the exceeded text leads to information loss and violates the Term Frequency Constrain (TF1). TF1 claims that replacing one of the non-query term word to a query term increases the relevance score. Therefore for a cut-offed document, if the replacement occurs in the exceeded part, the relevance score keeps the same in deep IR models.

For example, a document contains 5000 words in Figure 4, and the maximum length of our model set to 500 words. Thus we find that in the view of deep IR models, only few query terms “withdrawal” occur in the top 500 words of the document. While, as the figure shown, the query terms “methadone” and “baby” occur in the exceeded part of the document.

A more precision statistic of the last query term match position in a document is shown in Figure 5. The red line represents the 500 words threshold, thus about 40% of document loss the query term information because of the length limitation of the document. So this special preprocess affect 40% of the documents, which reflecting on the worse model performance.

Suggestions: For this situation, inspired by passage retrieval approaches, document can be split into several short passages. Then deep IR models apply on each $\langle \text{query}, \text{passage} \rangle$ pairs. Finally, the relevance score is the aggregation of each $\langle \text{query}, \text{passage} \rangle$ local relevance scores.

4.2.3 Embedding semantic abuse problem. Different from using one hot word representation in the hand-craft features such as BM25 and language model, recently most deep IR models adopt pre-trained word embeddings as the word representation, except the letter-trigrams used in DSSM and CDSSM. The advantage of adopting word embedding as the word representation is to investigate semantic matching information into the model. However, on the flip side, semantic matching brings too much noise matching signals, which covers up the exact matching signals and dominates the final matching score. Similar to the heuristic constrains proposed in [2] which only consider exact matching signals, we append one constrain by considering semantic matching signals, called Term Semantic Frequency Constrain (TSFC).

TSFC: Let $q = \{w\}$ be a query with only one term w . Assume $|d_1| = |d_2|$ and $s(w, d_1) = s(w, d_2)$. If $c(w, d_1) > c(w, d_2)$, then $f(d_1, q) > f(d_2, q)$.

This constrain assumes that two documents have the same length, and the sum of the semantic matching signals (including exact matching signals) are equivalent, $s(w, d_1) = s(w, d_2)$. The larger number of the exact matching signals $c(w, d)$, the higher relevance score $f(d, q)$.

Take an instance in LETOR 4.0 Dataset as an example shown in Figure 6. The given query term “noradrenaline” has a high similarity with so many medical related words, such as “epinephrine”, “catecholamine” and “metabolism”. Thus the sum of matching signals in the lower document is higher than that in the upper document, even the lower document dose not has any exact matching signal with query term “noradrenaline”. As the experiment shows that deep IR models prefer the lower document, for the sake of the higher density matching signals.

Suggestions: As TSFC shown that we need enlarge the gap between semantic matching signals and exact matching signals. As the Figure 7 shows that for interaction-focused models, we can define a proper similarity function between words, so that the exact matching signals are larger than all semantic matching signals, such as the similarity functions show in Figure 7(b) and Figure 7(c).

4.2.4 Feature Robustness. The robustness of the features can be interpreted as that when some of the features are missing, how much it affects the model performance. The features we take into consideration are the 46 dimensional features provided in LETOR 4.0 dataset and the last layer 20 dimensional outputs in the Arc-I and MatchPyramid. Then we use a linear model to fit these 3 sets of features to the final relevance labels. In this way, the learnt weights reflect the importance of the features.

In order to visualize the robustness different between hand-crafted features, Arc-I and MatchPyramid, we conduct experiments on the first fold of LETOR 4.0 dataset. After sufficient model training, we firstly sort features by its importance (their corresponding weight). Then remove features one-by-one following the order of the feature importance at each time. Finally, evaluate the performance of the rest of the features. The Figure 8 illustrates the procedure of the feature removing. Figure 8(a) shows the result of hand-crafted features, as we can see that when we remove the first two features the performance drops a lot to about 0.36. Figure 8(b-c) shows the results of Arc-I and MatchPyramid, as we can see that even half of the feature have been removed, the model performance affects a little. That is to say, deep IR models automatically learnt features turn to be more robustness than the hand-crafted features.

5 COMPARISONS AMONG DEEP IR MODELS

In this section, we investigate the differences between representation-focused models and interaction-focused models based on information retrieval task. Deep IR models lay into these two categories, such as DSSM, CDSSM and Arc-I belong to representation-focused models, Arc-II and MatchPyramid belong to interaction-focused models. We choose two classical models from each category, Arc-I and MatchPyramid. In order to explore their intrinsic differences, we conduct our analysis on Robust dataset, LETOR 4.0 dataset and a simulated dataset.

... library museum legend christmas fairy legend christmas fairy word fairy come latin word fata meaning fate means fairy cousin classical fates believe control fate destiny human race hope fairy associate christmas good ... good men germany christmas fairy legend tell ... patter small feet coming hall room open door fairy clad sparkle robe dance laugh singing splendid ... ernestine queen fairy came return lost gold ring love sight count otto soon ask fairy queen bride ... live happily years count otto fairy wife decided hunt forest near castle count grew impatient ...

... dr samuel harris national museum dentistry baltimore made possible generous support colgate palmolive children museum virginia select nine children science museum host branch bristle ... toothbrush ancient babylonia st century tooth sticks made wild ... proper space inspire visitor make healthy snack choice computer driven essential toothbrush tooth fairy tech savvy visitor option create virtual toothbrush learn evolutionary time periods emphasize dimension timeline visitor take time innovate design handcrafte toothbrush bench view vintage dental product poster find ...

Figure 3: Two documents related to query “tooth fairy museum”, the right document cover one more query term than the left document rank high in the deep IR models.

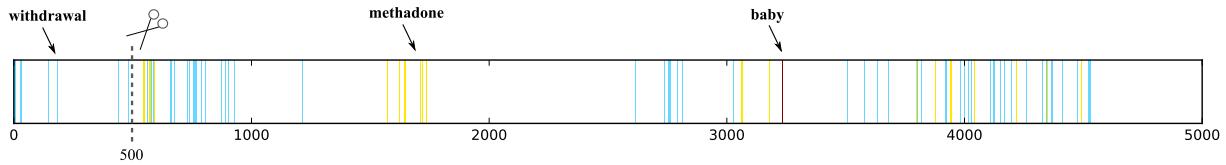


Figure 4: An example document shows that the exceed part of a long document has much query term matching information.

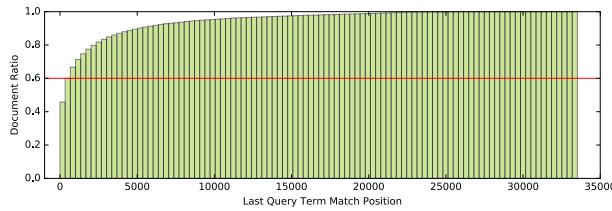


Figure 5: The distribution of last query term matching position.

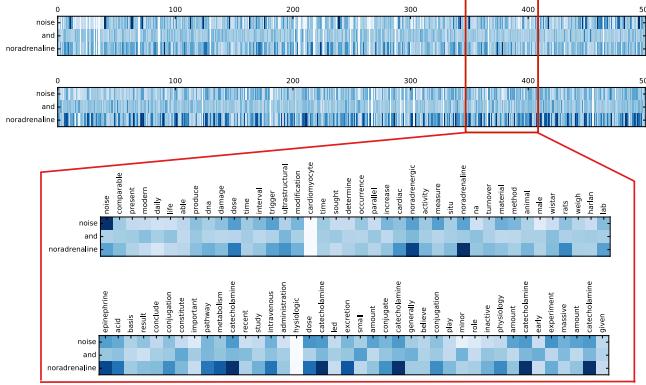
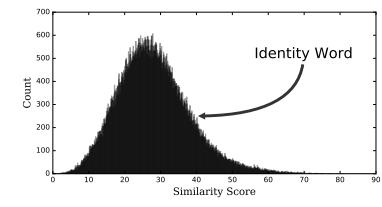
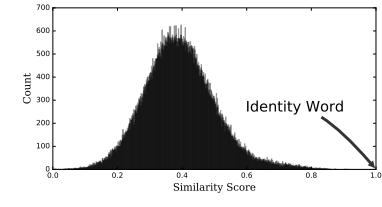


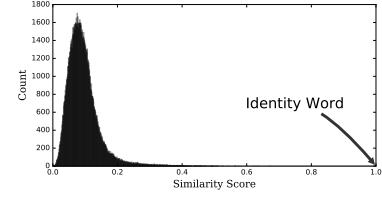
Figure 6: An example illustrates the embedding semantic matching problem. The first document contains exact matching signals, while the second document contains much high semantic matching signals.



(a) Dot Product



(b) Cosine



(c) Gaussian Kernel

Figure 7: Choose one word from the vocabulary and measure the similarity between other words, we draw the histogram of three type of similarity functions: dot product, cosine and gaussian kernel. The arrow point the similarity between two identity word (the word we choose).

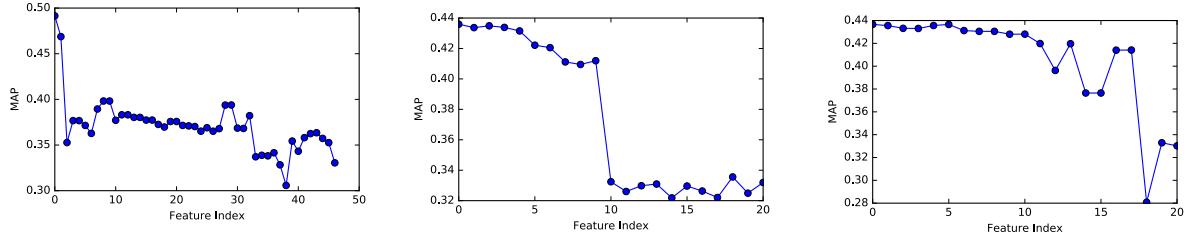


Figure 8: Left: 46 dimensional hand-crafted features; Middle: Arc-I last layer 20 dimensional features; Top: MatchPyramid last layer 20 dimensional features.

Table 3: Performance comparison of different deep IR models on Robust, LETOR 4.0.

Model	Robust				
	NDCG@1	NDCG@10	P@1	P@10	MAP
DSSM	0.122	0.137	0.122	0.135	0.048
CDSSM	0.118	0.134	0.118	0.130	0.042
ARC-I	0.124	0.138	0.124	0.132	0.050
ARC-II	0.140	0.148	0.140	0.156	0.054
MATCHPYRAMID	0.364	0.242	0.364	0.240	0.164
Model	LETOR 4.0				
	NDCG@1	NDCG@10	P@1	P@10	MAP
DSSM	0.290	0.371	0.345	0.352	0.409
CDSSM	0.288	0.325	0.333	0.291	0.364
ARC-I	0.310	0.386	0.376	0.364	0.417
ARC-II	0.317	0.390	0.379	0.366	0.421
MATCHPYRAMID	0.362	0.409	0.428	0.371	0.434

5.1 Performance Comparison

The performance comparison results illustrate in Table. 3. After comparing different models on different datasets, we can conclude as follow: 1) interaction-focused models are performed better than representation-focused models; 2) the gap between representation-focused models and interaction-focused models on dataset Robust is larger than the gap on dataset LETOR 4.0.

It motivates us to explore the differences between representation-focused models and interaction-focused models.

5.2 Property Analysis

5.2.1 Text Representations. Text representations is a major task for representation-focused models, since text representation is used to compress most valuable and distinguishable information into one vector. In paraphrase identification task, two sentences are symmetric and have similar length. However, in information retrieval, query and document are totally different objects. Query is abstract and almost every terms in it reflect a perspective of search intent. Thus, text representation for a query need to keep all the query terms information in one vector. On contrast, document is elaborate and relevant document just follow two hypotheses in the literature [11]. The Verbosity Hypothesis assumes that a long document is like a short document, covering a similar scope but with more words; while the Scope Hypothesis assumes a long document

Query	Pooling	Topic48	Topic36	Topic37
sante fe new mexico	national park office www two ⋮	1. park 2. national 3. nm 4. lake 5. area ⋮	1. federal ⋮ 21. service 22. office 23. issue ⋮	1. farm ⋮ 33. labor 34. www 35. urban ⋮

Figure 9: The Arc-I pooling words have relation to LDA topic words.

consists of a number of unrelated short documents concatenated together. Thus, text representation for a document only need to contain the important part of a document.

In order to analyze the information encoded in text representation, we visualize the words at the max pooling position in Arc-I model.

As an example, in Figure 9, the query terms are “**sante fe new mexico**”, which are listed in the first text box. The words selected by max pooling layer based on convolution output feature maps, are listed in second text box, such as “national” “part”. It is evident to see, all of the pooling words are not the words appear in query terms. The reason of the above observation is that the document and query text representations are extracted independently, thus the relation of these two representations are weak. For document, most informative words are extracted to composite the representation, which have the same purpose with the topic models. Thus we assume that the pooling words are related to the topic words in the topic models.

To further check this assumption, we conduct comparison with topic model. Without loss of generality, we choose the state-of-the-art, LDA model [1] to generate topic words. LDA model is trained on the whole corpus, with 50 topics and default parameters in package GENSIM [10]. Then we collect top 50 words in each topic, totally 853 words (some words lay in multiple topics). Meanwhile, we collect all pooling words using the Arc-I model, and sort decreasing by the words frequency. The Figure 10 shows the distribution of the word overlap ratio between top pooling words and the top 50 LDA topic words. From the results as we can see, 80% of top 500 pooling words come from the top 50 LDA topic words, for example in Figure 9, “national” and “park” come from the topic 48 and “office”

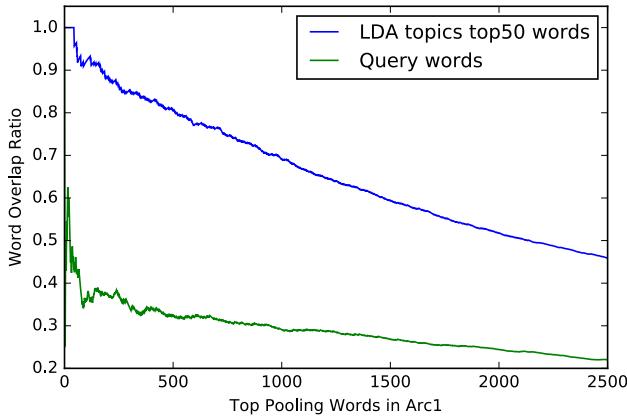


Figure 10: The overlap ratio of top pooling words in Arc-I with LDA topics top50 words and query words.

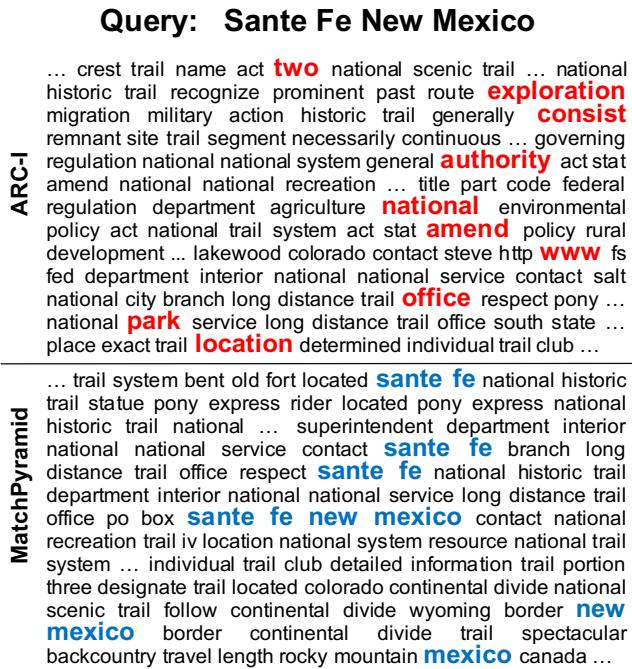


Figure 11: The different pooling words, highlighted using bold font, in the model of Arc-I and MatchPyramid.

come from the topic 36. Additionally, we also measure the overlap ratio between top pooling words and query words (green line in Figure 10). Contrast with LDA topic words, the overlap ratio of query words is relative small, for example about 30% of top 500 pooling words come from the query terms.

With above observation, we can conclude that representation-focused models, eg. Arc-I, generate topic related text representation as the topic model do.

5.2.2 Interaction Representations. Interaction-focused models aim to extract interaction representations on the top of interaction matrix, which is constructed using word-by-word similarity. In paraphrase identification, interaction representation can be interpreted as a hierarchical matching signal composition process, as described in [8]. Firstly, word level matching signals can be composited to phrase level matching signals, for example n-grams matching and n-terms matching. Then, conduct several composition steps to achieve sentence level matching signal. However, it differs a lot when we apply MatchPyramid in information retrieval task. Query is too short to be treated as a sentence, for instance, query “tooth fairy museum” only can be treated as a phrase or query “guatemala” only has one word. So that hierarchical matching signal composition reduces to word/phrase matching signals aggregation.

In order to understand what interaction-focused models have learnt, we analyze the MatchPyramid model from the pooling words. Different from the Arc-I model, given a query document pair, almost all pooling words of MatchPyramid come from the query terms. Figure 11 shows the different words pooling by Arc-I and MatchPyramid. In MatchPyramid, we use cosine similarity function to evaluate words similarity, thus identical words in query and document achieve the highest similarity value as 1. Then the following convolutional and pooling operations have a high probability to pool out these words.

5.2.3 Synthetic Analysis. From the above discussion, we realize that presentation-focused models and interaction-focused models capture different kinds of informations in IR task. In order to distinguish the performance of text representations and interaction representations, we introduce an additional synthetic dataset to verify the ability of two kinds of deep IR models.

The synthetic dataset is constructed using two well-designed ground truth.

- (1) **Topic Match** The relevance document contain a specific topic, where each topic expresses as a sequential of words. For example, word sequence “neural network” represent a topic, and any document contains this word sequence lays into this topic, marked as relevance.
- (2) **Density Match** The relevance degree of a document is proportional to the density of query terms it contained. For example, the document contain 100 query terms is more relevant compare to the one contain 10 query terms.

The whole dataset contains 10,000 queries. In each query, words are randomly sampled from a 2000 words vocabulary. Then, for each query we construct one relevant document and four irrelevant documents. In each document, words are randomly sampled from the same vocabulary with query. The length of query is ranging from 2 to 8, while the length of the document is ranging from 300 to 700.

Arc-I and MatchPyramid are evaluated on these two dataset respectively, and the results are shown in Figure 12. As we can see, the performance of representation-focused model Arc-I and interaction-focused model MatchPyramid is quite the opposite. The representation-focused model have ability to learn topic information, while very little information about query terms density. On

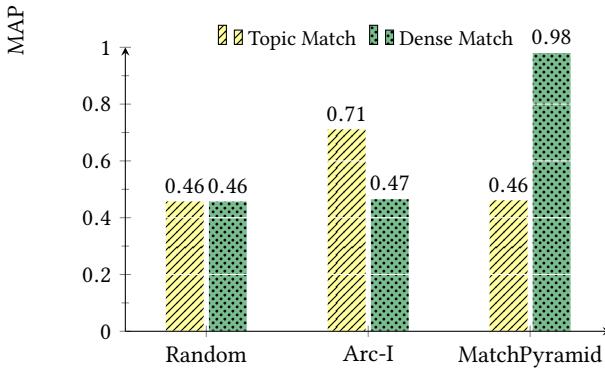


Figure 12: Performance comparison of Random, Arc-I and MatchPyramid on synthetic data.

the contrast, interaction-focused model is good at learning query terms density information, but almost no topic information covered.

6 CONCLUSIONS AND FUTURE WORK

As a conclusion, deep IR models still perform worse than hand-crafted features. The possible reason is that under a limit size of dataset, hand-crafted features, such as BM25, obey the heuristic retrieval constrains, while deep IR models ignored. Apart from using larger dataset, we establish guidelines to explicit using heuristic retrieval constrains, in order to further improvement of deep IR models.

On comparing representation-focused models and interaction-focused models, we can conclude that, 1) representation-focused models focus on learning good text representation, which encode the topic related words in the final representation; 2) interaction-focused models focus on learning good interaction representation, which good at collecting the density of matching signals. These interesting findings pave a way to better understand the different deep IR models and demonstrate that for different applications we need to choose different deep IR models.

REFERENCES

- [1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [2] Hui Fang, Tao Tao, and ChengXiang Zhai. 2004. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 49–56.
- [3] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM*. ACM, 55–64.
- [4] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*. 2042–2050.
- [5] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. ACM, 2333–2338.
- [6] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.
- [7] Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*. 1367–1375.
- [8] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *AAAI*. AAAI Press, 2793–2799.
- [9] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. 2010. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval* 13, 4 (2010), 346–374.
- [10] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. <http://is.muni.cz/publication/884893/en>.
- [11] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*. Springer-Verlag New York, Inc., 232–241.
- [12] Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval. (1986).
- [13] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *WWW*. International WWW Conferences Steering Committee, 373–374.