Empirical Study of Multi-level Convolution Models for IR Based on Representations and Interactions

Yifan Nie
Université de Montréal
Montréal, Québec, Canada
yifan.nie@umontreal.ca

Yanling Li
College of Computer and Information Engineering, Inner Mongolia Normal University
Hohhot, Inner Mongolia, China
cieclyl@imnu.edu.cn

Jian-Yun Nie
Université de Montréal
Montréal, Québec, Canada
nie@iro.umontreal.ca

ABSTRACT

Deep learning models have been employed to perform IR tasks and have shown competitive results. Depending on the structure of the models, previous deep IR models could be roughly divided into: representation-based models and interaction-based models. A number of experiments have been conducted to test these models, but often under different conditions, making it difficult to draw a clear conclusion on their comparison. In order to compare the two learning schemas for ad hoc search under the same condition, we build similar convolution networks to learn either representations or interaction patterns between document and query and test them on the same test collection. In addition, we also propose multi-level matching models to cope with various types of query, rather than the existing single-level matching. Our experiments show that interaction-based approach generally performs better than representation-based approach, and multi-level matching performs better than single-level matching. We will provide some possible explanations to these observations.

CCS CONCEPTS

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

KEYWORDS

Information Retrieval; Neural Network; Ranking

ACM Reference Format:
Yifan Nie, Yanling Li, and Jian-Yun Nie. 2018. Empirical Study of Multi-level Convolution Models for IR Based on Representations and Interactions. In *ICTIR '18: 2018 ACM SIGIR International Conference on the Theory of Information Retrieval, September 14–17, 2018, Tianjin, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3234944.3234954

1 INTRODUCTION

Deep learning techniques have been successfully used first in image [7] and speech processing [2]. The key idea behind them is to learn representations to represent the content and features of images and speech. The main architecture is convolutional neural network (CNN), which aggregates representation cells at a lower-level to form a higher-level representation. It has been observed that the representations can successfully capture features from lines, forms, face components, to specific face classes (persons) at different levels in an CNN trained for face recognition. The CNN architecture demonstrates high capability of creating more and more complex and abstract features when we move up in the convolution layers.

These techniques have then been extended to text processing. To cope with the specificities of texts, deep learning techniques have been much extended, namely to incorporate the sequential dependencies between words in texts. In particular, recurrent neural networks (RNN) are widely used for different tasks in text processing: machine translation, question-answering, and so on.

The great success of deep learning has triggered a tremendous interest in the IR community. A large number of research papers on neural IR models are proposed, which are based on CNN or RNN. It has been shown that RNN can be successfully used in tasks that deal with short texts such as in question-answering (i.e. to re-rank short answers) [11, 14]. However, RNN has not been often used for long texts, in which RNN has difficulty to capture the essential part of a long text. For the core ad hoc IR task, CNN is the main architecture used in most of the previous studies on neural IR models.

The previous studies have proposed two main families of models: representation-based models and interaction-based models. Representation-based models such as DSSM [5], CDSSM [12] focus on learning meaningful representations through several hidden layers and apply a similarity function on the last level query and document representations to estimate relevance. Instead of learning semantic representation of query and documents, interaction-based models calculate local interactions of each query and document term at input and learn the term-level interaction patterns through several hidden layers. In previous work, the two approaches have been tested under different experimental conditions, making it difficult to compare them fairly. The goal of this paper is to make a fair comparison of the main models proposed for ad hoc search. The first question we examine is which of the representation-based and interaction-based models, when implemented in a similar manner, better suit ad hoc IR tasks.

Apart from the above issue, we also observe that previous neural IR models only employ the representations or interaction scores of the last level to produce a global interaction score. However, users queries may be of different nature. For example, the query "fact on Uranus" (a ClueWeb query) is a lexical query for which

ICTIR '18, September 14–17, 2018, Tianjin, China

ACM ISBN 978-1-4503-5656-5/18/09…$15.00
https://doi.org/10.1145/3234944.3234954

a low-level exact match for the word "Uranus" is required. A semantic matching may run the risk of matching with other planets, leading to query drift. On the other hand, the query "last supper painting" is a conceptual query which needs some generalized representations/interactions on higher level of the model. These queries require different levels of matching, from lexical to semantic levels. This is the general case for IR: users may submit queries to locate documents containing the same words, or the same concepts. In order to cope with this issue, we extend the existing representation- and interaction-based models to multi-level matching.

We will run extensive experiments under the same test condition to compare these models. Our results will clearly show that interaction-based models are usually preferred to representation-based models for ad hoc search; and multi-level matching is preferred to single-level matching.

2 RELATED WORK

In deep IR models, given the query q and document d, the matching is often achieved by estimating a relevance score rel of q and d. Depending on how to produce the relevance score $S(q, d)$, previous deep IR models could be roughly divided into 2 categories: representation-based models [5, 9, 12] and interaction-based models [3, 4, 10].

Representation-based models focus on learning meaningful semantic representations through several hidden layers and estimate a global relevance score by a matching function applied on the last level representations of the query and document. The process could be summarized in Equation 1.

$$rel(q, d) = S(\phi(q), \phi(d)) \tag{1}$$

where ϕ is a complex feature function to map query or document text into meaningful semantic representations through several hidden layers. S is a matching function, such as cosine or dot similarity. For example, in DSSM [5], the feature function ϕ is a feed forward neural network and S is a cosine similarity. In CDSSM [12], ϕ is a convolutional network and S is the cosine similarity.

Different from representation-based models, interaction-based models focus on learning salient interaction patterns from the input local interactions through a series of hidden layers. The process could be summarized in Equation 2.

$$rel(q, d) = S_n \circ S_{n-1} \circ \ldots \circ S_0(w(q), w(d)) \tag{2}$$

where w is often a simple embedding lookup function which will extract word embeddings of corresponding terms, and the matching function is a composition of a series hidden neural transformations $S_n \circ S_{n-1} \circ \ldots \circ S_0$.

For example, in MatchPyramid [10] and ARC-II [4], the feature function w maps each term of query and document into word embedding vector, and the matching function $S_n \circ S_{n-1} \circ \ldots \circ S_0$ is a deep convolutional network of several layers to learn the matching patterns from the input interaction matrix.

Similarly, the DRMM model [3] calculates the interactions between each query term with each document term by a similarity function, and the histogram of interactions between query term t_i^q and all document terms are produced. Afterwards, n weight-sharing feed-forward neural networks take the histograms as input and predict n matching scores, where n is the length of the query. Finally the n scores are aggregated through an aggregating gate to produce a global matching score.

The two families of approaches have been extensively tested. [5] and [12] showed that representation-based models trained on click-through data could successfully produce superior effectiveness than a traditional model (BM25) on document title retrieval. However, [10] and [3] showed that DSSM and CDSSM were far less effective than BM25 on ad hoc retrieval and interaction-based models performed better. It is difficult to draw a solid conclusion from these experiments because the retrieval tasks and test conditions are very different. This is the very motivation of our paper - to compare the two approaches under the same test condition. In this study, in order to understand the contribution of representation-based and interaction-based models, we test them separately in this paper.

The complementary effects of representation-based and interact ion-based models have been observed. Therefore, they are combined in some models. For example, the DuetNet [8] incorporates the strengths of both representation-based and interaction-based models by explicitly building 2 sub-models. In this paper, however, we intend to compare the two learning and matching schemas directly, without mixing up other aspects. So, we do not consider such a combined model in this study.

In general, a neural IR model tends to create a high-level representation or matching pattern along the convolution layers. It has been noticed that such a model may fail to deal with lexical queries. To address this issue, the AttR-Duet model [13] exploits both word and entity matching features like BM25 and TF-IDF scores and builds two 1D CNN models to produce 2 matching scores, and the word and entity matching scores are linearly combined with attention weights learned by a separate attention model. [13] shows the importance of low-level lexical features. In our study, we will also combine lexical and semantic matchings, but in a different architecture. We propose to combine the matching scores at multiple levels of convolution.

Multiple matchings have also been considered in MultiGranCNN [15], which allow two word sequences to match at different granularities: word, phrase and sentence. However, the model has only been tested in phrase matching tasks.

From the above analysis, we can observe that previous neural IR models only employed the final level or low-level representation/interaction score to estimate a global relevance score. However, user's queries may be of different nature, which may require matching at different levels of abstraction. Therefore we propose to investigate the possibility of integrating multi-level matching into representation-based and interaction-based models. The details will be discussed in Section 5.

3 DATASET AND EXPERIMENTAL SETTINGS

Before presenting the details of different models, we first describe the experimental conditions: the test collection, the training method and some general settings of the neural models. These settings will be shared by all the models tested. Some details provided in this section may become clearer when the models are described.

Experiments are conducted on the ClueWeb09B collection. The detailed statistics of the dataset are summarized in Table 1. We choose to test on this collection because it is one of the most difficult

test collections for ad hoc search, and is closely related to web search. The test queries are #51-200, while queries #1-50 are used as validation queries for hyper-parameter tuning.

Table 1: Collection Statictics

Collection	Genre	Validation Queries	Test Queries	#Docs	Avg.d.length
Clueweb09B	Webpages	1-50	51-200	50M	1,506

A critical aspect in neural model training is the need of a large amount of training data. Different training data have been used in the previous experiments, leading to inconsistent experimental results. For example, Huang et al. [5] and Shen et al. [12] showed that DSSM and CDSSM could be successfully trained on clickthrough data, and they outperformed BM25 for title retrieval on their proprietary test set. However, Guo et al. [3] showed that the same models trained using limited amount of data (true relevance judgments) led to effectiveness far below that of BM25. To be fair, different models should be trained using the same training and test data. Ideally, we should use a large amount of manually annotated data or clickthrough data. However, such data are not publicly available. Therefore, we sort to weak supervision using a traditional model (BM25 in our case). This weak supervision has been found to be able to train a neural model reasonably, enabling the trained neural model to outperform BM25 [1]. To generate weak supervision labels, we employ the AOL query logs[1] and filter out navigational queries[2] and queries containing non-alphanumeric characters as done in [1]. This results in 8,969,337 training queries. We retrieve the top 50 documents using Indri[3] BM25 model with default parameters ($k_1 = 1.2, b = 0.75, k_3 = 1000$) and convert the results to positive and negative training examples as follows: For a given query, we randomly sample 2 documents and regard the one with higher BM25 score as positive document, the other one as negative document. We employ a pair-wise training scheme. The loss is defined in Equation 3, where $S(Q, D_+)$ and $S(Q, D_-)$ are the predicted scores for positive and negative example, Θ includes all trainable parameters of the model.

$$L(Q, D_+, D_-; \Theta) = max(0, 1 - (S(Q, D_+) - S(Q, D_-))) \quad (3)$$

For each validation and test query, we return top 1000 documents by BM25 model as candidate documents and use our model to rerank them by the inferred matching score.

Let us specify the general setting of our experiments applied to all the tests, even though the models will only be described later. We set the max query length and document length to be $n = 15$, $m = 1000$ and apply zero paddings as done in [10]. The maximum query length limit is enough to cover most of the queries in the training set and all the queries in the validation and test sets (4 and 5 words respectively). For documents, the parts that exceed the limit are cut off. We employ pre-trained GloVe.6B.300d embeddings[4] similar to [1]. It would be possible to train word embeddings from scratch. We will leave this to future work. We fix the embeddings for interaction-based models and continue to fine-tune them during training for representation-based models, and omit OOV document terms.

We employ the mean average precision (MAP) [16] and nDCG [6] as evaluation metrics. We perform paired t-test with respect to the BM25 baseline, and the statistically significant results are marked with * in the result tables.

4 REPRESENTATION-BASED VS INTERACTION-BASED MODELS

In this experiments, we will compare the typical representation- and interaction-based models proposed in the literature. There are many variants in the models. The models we test here capture the essence of most of the existing models, and we focus on the representation- vs. interaction-based learning.

Representation-based Model: The architecture of the Representation-based Convolutional model is presented in Fig 1. This model

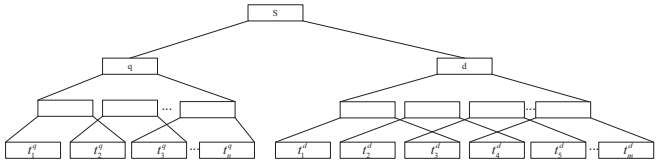


Figure 1: Representation-based Convolutional Model

is similar to CDSSM [12] without the word transformation to letter-trigram. In this model, the query terms and document terms are combined into more abstract representations through 1D convolutions [17].

Let q and d denote the query and document respectively. In this model, q and d are represented by a sequence of word embeddings $q = [t_1^q, t_2^q, ..., t_n^q]$ and $d = [t_1^d, t_2^d, ..., t_m^d]$, where t_i^a and t_j^d represent the word embedding of the i^{th} query term and the j^{th} document term respectively. Then, a series of 1D convolutions are performed to combine the word representations into more abstract representations as follows.

$$C_i^{q,(1)} = f(W_1^q * [t_{i-w(1)}^q; ..; t_{i+w(1)}^q] + b_1^q) \quad (4)$$

$$C_i^{q,(k)} = f(W_k^q * [C_{i-w(k)}^{q,(k-1)}; ...; C_{i+w(k)}^{q,(k-1)}] + b_k^q),\ k = 2, ..., L \quad (5)$$

$$C_i^{d,(1)} = f(W_1^d * [t_{i-w(1)}^d; ..; t_{i+w(1)}^d] + b_1^d) \quad (6)$$

$$C_i^{d,(k)} = f(W_k^d * [C_{i-w(k)}^{d,(k-1)}; ...; C_{i+w(k)}^{d,(k-1)}] + b_k^d),\ k = 2, ..., L \quad (7)$$

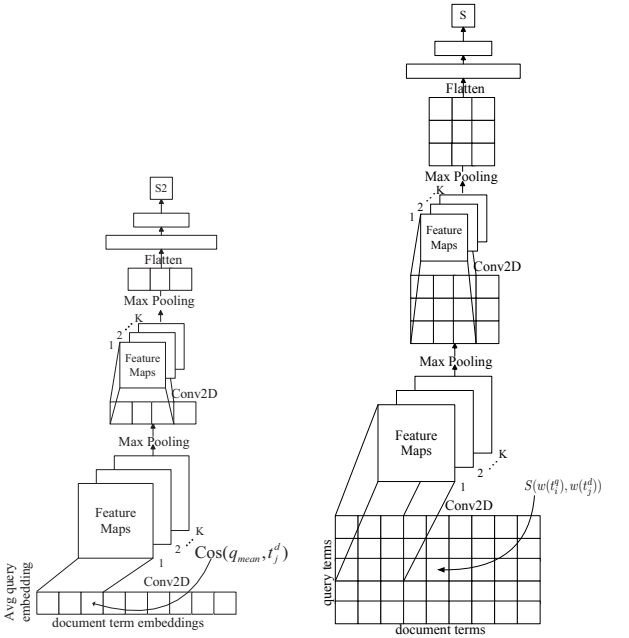
where W_k^q, b_k^q, W_k^d, b_k^d are the weight and bias for the query and document of the k^{th} layer respectively; t represents the input word embedding layer and $C_i^{q,(k)}$, $C_i^{d,(k)}$ are the i^{th} convolved vectors of the k^{th} layer; $2w(k) + 1$ is the window size for the k^{th} layer; f is a non-linear transformation. Once the final level representations of the query $C^{q,(L)}$ and the document $C^{d,(L)}$ are obtained, a cosine similarity function is applied to estimate the global matching score as follows.

$$S_L = Cos(C^{q,(L)}, C^{d,(L)}) \quad (8)$$

[1] http://octopus.inf.utfsm.cl/~juan/datasets/
[2] Queries containing URL strings ("www.", ".com", ".org", ".net", ".edu")
[3] https://www.lemurproject.org/indri.php
[4] https://nlp.stanford.edu/projects/glove/

During experiments, we limit the max number of convolution layers L to 3 and fix the hidden size to be 128. For 2-convolutional-layered model, the convolutional window sizes are set to [3, 13] for query side (i.e. 3 for the first layer and 13 for the second layer), [3, 998] for document side, and all strides are set to 1. For 3-convolutional-layered model, the convolutional window sizes are set to [3, 5, 9] and strides are set to [1, 1, 1] for query side, and [3, 10, 198] and strides are set to [1, 5, 1] for document side. Notice that we have tested many other settings for this and other models, but the ones described in the paper tend to produce the best results. Therefore, we will omit the other settings in this paper.

Interaction-based Model: The interaction-based model we implement is inspired by MatchPyramid, which has several convolution and pooling layers on top of the basic interaction matrix between document and query terms [10]. This architecture represents the essence of the family of interaction-based models (although there are some quite important details in other alternative models). The architecture of the Interaction-based Convolutional Model is presented in Fig 2.



(a) 1D Inter **(b) 2D Inter**

Figure 2: Intercation-based Convolutional Model

At input, an interaction grid I is constructed. Afterwards, several convolutional layers and max-pooling layers [10] are constructed to learn the underlying interaction patterns. Finally, a MLP is added on top of the last max-pooling layer to extract a relevance score as global matching score. There are 2 ways to build the interaction grid I: Either we build the interaction between the global query representation and each document term representation (1D interaction) or we build the interaction between each query term and document term (2D interaction). In this study, both 1D and 2D interactions are explored.

In 1D interaction-based model, the interaction I reduces to a vector. We first calculate a global query representation by taking the average of all query term embeddings as follows.

$$q_{mean} = \frac{1}{n}\sum_{i=1}^{n} t_i^q \tag{9}$$

where n is the query length, and t_i^q is the embedding vector of the i^{th} query term.

Then the interaction vector I is constructed by calculating the cosine similarity between q_{mean} and each document term embedding t_j^d as follows.

$$I_j = cos(q_{mean}, t_j^d) \tag{10}$$

During experiments of the 1D interaction-based model, the convolution filter sizes are set to [3, 5] for the model with 2 convolutional layers and [3, 5, 9] for the model with 3 convolutional layers. The pooling size is set to 2 for each max-pooling layer of both models. The number of feature maps is set to 128 for both models.

In 2D interaction-based model, the interaction I is a matrix with each entry I_{ij} being the cosine similarity of query term t_i^q and document t_j^d calculated as follows.

$$I_{ij} = Cos(t_i^q, t_j^d) \tag{11}$$

During experiments, we limit the max number of convolution layers L to be 2 due to memory limit and fix the number of feature maps to [32, 16] for the 2 convolution layers. We fix the pooling size of all max pooling layers to be (2, 2). The filter shapes of the 2 convolutional layers are fixed to (3, 3) and (5, 5) because they produced good results in our preliminary study.

Once the interaction grid I is constructed, a series of convolutions and max-poolings are performed as follows.

$$C_1^k = f(W_1^k * I + b_1^k),\ k = 1, .., K \tag{12}$$

$$P_1^k = max\_pool(C_1^k),\ k = 1, .., K \tag{13}$$

$$C_i^k = f(W_i^k * P_{i-1} + b_i^k),\ i = 2, .., L,\ k = 1, .., K \tag{14}$$

$$P_i^k = max\_pool(C_i^k),\ i = 2, .., L,\ k = 1, .., K \tag{15}$$

where C_i^k is the feature map k of the i^{th} convolved layer; I is the input interaction matrix; W_i^k and b_i^k are the kernel and bias of layer i for the feature map k; L is the number of convolution layers, and K is the number of feature maps; f is a non-linear mapping; and $*$ represents the convolution operator.

In order to determine the global matching score, the last max-pooled layer is flattened into a 1D vector and fed into a fully connected MLP to output a scalar score S

$$h = g(W^h P_L + b^h) \tag{16}$$

$$S = h(W^s h + b^s) \tag{17}$$

where h and S represent the hidden layer of the MLP and the matching score respectively; W^h, b^h, W^s, b^s are the weights and biases for the hidden and scoring layer; g and h are non-linear mappings.

The experimental results are presented in Table 2, where Rep-2L, Rep-3L, Inter-1D-2L, Inter-1D-3L and Inter-2D-2L represent the representation-based models with 2 and 3 convolution layers, 1D

interaction models with 2 and 3 convolution layers and 2D interaction model with 2 convolution layers respectively. From Table 2, we

Table 2: Results of Representation-based and Interaction-based Models[1]

Model	MAP	NDCG@1	NDCG@3	NDCG@10	NDCG@20
BM25	0.0879	0.1178	0.1359	0.1356	0.1394
Rep-2L	0.0121	0.0019*	0.0024	0.0042	0.0053
Rep-3L	0.0145*	0.0158*	0.0143*	0.0143*	0.0149*
Inter-1D-2L	0.0663*	0.0927	0.0846*	0.0912*	0.0967*
Inter-1D-3L	0.0632*	0.0662*	0.0922*	0.0904*	0.0957*
Inter-2D-2L	**0.0884**	**0.1485***	**0.1423**	**0.1411**	0.1389

observe that representation-based models don't perform well. One possible reason is that it is difficult to learn good global representation of the document which is often very long. Since the model employs only the final level representations to estimate relevance, the performance will be influenced by the quality of the global representations. Intuitively, it is very difficult, even impossible, to represent every aspect of a long document in a single vector, and that the vector should be appropriate to match with any related query. This may be too much to ask. The poor effectiveness of representation-based models has been observed in several previous studies [3, 10]. Our observation is consistent. However, this result is inconsistent with that of [1]. Dehghani et al. [1] showed that a neural model based on representation learning, weakly supervised by BM25, can lead to performance superior to BM25. It is difficult to understand our failure. Our explanation lies in the huge difference in computation resources: In [1], a very large number of epochs (10^7) have been used in training, while we can only afford a limited number of epochs in our experiments (and in a real situation). In Fig. 3, we can see that the effectiveness of representation-based models stagnates on validation queries along the number of epochs, although we cannot exclude the possibility that they become close to BM25 after many millions of epochs.

We also observe that the Inter-2D-2L model could get competitive result with the BM25 model with the help of weak supervised data (see also Fig. 3 on validation queries). The sharp contrast between interaction-based and representation-based models suggests that the former can better capture useful matching signals than the latter. In contrast to a global representation for a document, the interaction-based models try to determine local matching signals between document and query. Local matching signals are known to be important in IR - in fact, all the traditional models are built on similar local matching signals. This also correspond to our understanding of the general search tasks: a document is relevant often because parts of its content are relevant. The local matching signals reflect this very principle. This observation has also been made in some previous studies (e.g. [3]).

It could be possible that the difference between the Inter-2D model and the Rep models is due to the use of 1D and 2D convolutions. To better understand this aspect, we compare Inter-1D with Rep models, which all use 1D convolution. In Fig. 3, we can see that Inter-1D models (with 1 or 2 layers) clearly outperform representation-based models. This observation shows that the main difference is due to the learning object - representation or interaction pattern. However, we also observe that Inter-1D models have lower effectiveness than Inter-2D model. This shows that the use of 2D convolution on the basis of term interactions may capture better matching signals than the 1D convolution based on the interactions with the whole query. The granularity of the interactions matters.

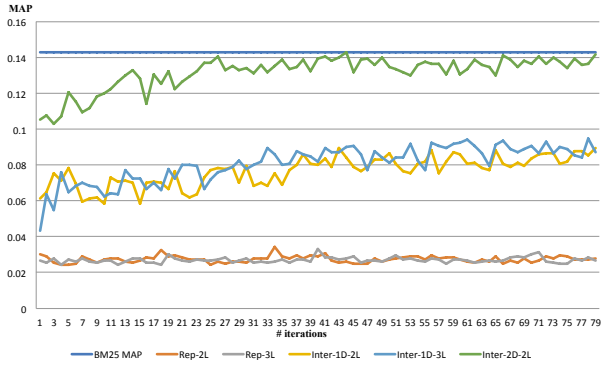


Figure 3: Validation Curve on MAP

From Fig 3, we can observe that for the Inter-2L, at the beginning of the training, the model performance is lower than the MAP of BM25 model. As training goes on, the performance would increase and finally approach the MAP of BM25. This observation confirms the effectiveness of weak supervision. However, for the representation-based models (Rep-2L and Rep-3L), the performance could not catch up with BM25, showing that learning representation is more difficult than learning interaction patterns.

5 MULTI-LEVEL MATCHING REPRESENTATION-BASED VS INTERACTION-BASED MODELS

As we discussed earlier, user's queries have very different natures and the required matching can range from lexical matching to more abstract semantic matching. Therefore, a neural IR model should contain several levels of matching between query and document. In this section, we will extend the representation- and interaction-based models to incorporate multiple levels of matching.

Representation-based Multi-level Matching Model:

The Representation-based Multi-level Matching Model (Multi-Match-Rep) is an extension from the representation-based model described in the previous section, in which we will define a matching function at every level of representation. The architecture is shown in Fig 4. Notice that we define a matching score S0 at the very basic level of word representation in order to capture some signals of term matching. It is possible to replace this function by a traditional matching function such as BM25, as in [11], but, as we explained, we want to keep the whole architecture within the neural framework in this study. The other matching functions at higher levels are intended to capture matching at more abstract and semantic levels.

The matching scores at different levels are determined as follows: For the i^{th} level, we construct a interaction matrix between the

[1]* means statistically significant difference with BM25

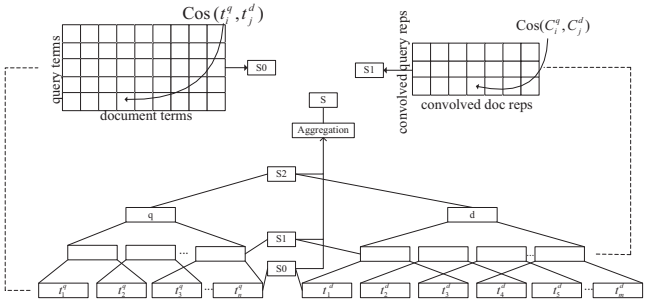


Figure 4: Representation-based Multi-level Matching Model

query and document representation vectors.

$$I_{ij}^{(0)} = Cos(t_i^q, t_j^d) \tag{18}$$

$$I_{ij}^{(k)} = Cos(C_i^{q,(k)}, C_j^{d,(k)}),\ k = 1, .., L-1 \tag{19}$$

where I^k is the interaction matrix of the k^{th} level. t_i^q and t_j^d are the word embeddings of the i^{th} query term and the j^{th} document term. $C_i^{q,k}$ and $C_j^{d,k}$ are the i^{th} convolved vector for the query at and the j^{th} convolved vector for the document at layer layer k respectively.

Once the interaction matrices are produced, a series of level-specific scores are extracted as follows: We take the top P interactions across each row u of $I^{(k)}$ (P is set to 5 in this study) and average them to obtain a scalar value $M_u^{(k)}$. This $M_u^{(k)}$ represents the strongest interactions between the u^{th} query representation and the document. The idea behind is that the matching score only depends on a few matching spots in the document instead of the entire document. We then sum up $M_u^{(k)}$ for every query term/representation to obtain a matching score $S^{(k)}$ for the level k. The final level matching score $S^{(L)}$ is estimated by the cosine similarity of the global query and document representation. The process could be summarized as follows.

$$M_u^{(k)} = \frac{1}{P}\sum_{p=1}^{P} \underset{v=1..m}{\text{top P}}\ I_{uv}^{(k)},\ \ S^{(k)} = \sum_{u=1}^{n} M_u^{(k)} \tag{20}$$

Once the scores of each level are extracted, they are aggregated through a softmax gate to produce the global matching score S as follows.

$$\beta_k = \frac{exp(\alpha_k S^{(k)})}{exp(\sum_{k=0}^{L} \alpha_k S^{(k)})},\ \ S = \sum_{k=0}^{L} \beta_k S^{(k)} \tag{21}$$

where α_i are learnable parameters, $S^{(k)}$ is the matching score of the level k.

Interaction-based Multi-level Matching Model:

The Interaction-based Multi-level Matching Model (Multi-Match-Inter) is presented in Fig 5. The convolution-pooling part (left part) of the model is identical to that described in the previous section. What is added is a series of matching scores S_i at every level of convolution, as well as an aggregation layer to combine these scores into a global score. We provide details about them.

For the input interaction matrix I and each convolved layer C_i, a scalar feature $M^{(i)}$ will be calculated. For the input interaction

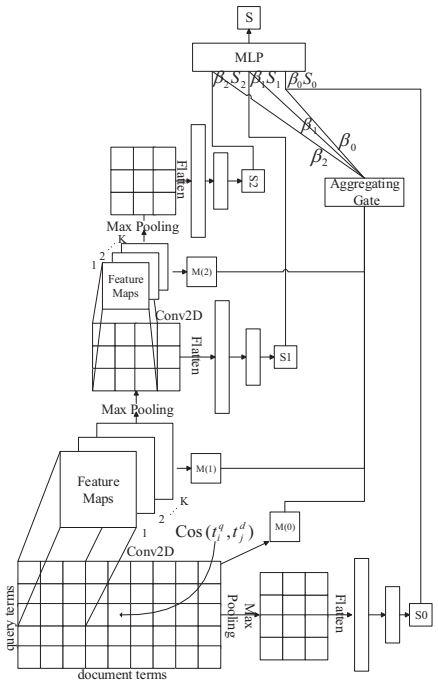


Figure 5: Interaction-based Multi-level Matching Model

matrix I, we take the max interaction values $M_u^{(0)}$ across each row u, which represents the max matching intensity across all document terms for the query term $t_u^{(q)}$. Afterwards, we sum up all the $M_u^{(0)}$s for each query term $t_u^{(q)}$ and get the global maximum interaction value $M^{(0)}$ for the whole query with respect to the document. This quantity reflects the word-level matching between document and query. The process could be summarized as follows.

$$M_u^{(0)} = \max_{v=1..m} I_{uv},\ \ M^{(0)} = \sum_{u=1}^{n} M_u^{(0)} \tag{22}$$

For each feature map $C_i^{(k)}$ in the convolved layer i, we proceed in the same way to obtain a $M_{(k)}^{(i)}$ for this specific feature map k. Then we average the $M_{(k)}^{(i)}$s to obtain the global $M^{(i)}$ for this convolution layer. The process could be summarized as follows.

$$M_{u,(k)}^{i} = \max_{v=1..m} [C_i^{(k)}]_{uv},\ u = 1, .., n,\ k = 1, .., K \tag{23}$$

$$M_{(k)}^{(i)} = \sum_{u=1}^{n} M_{u,(k)}^{(i)},\ \ M^{(i)} = \frac{1}{K}\sum_{k=1}^{K} M_{(k)}^{(i)} \tag{24}$$

The motivation of designing the M features is to try to capture the importance of each interaction level. Intuitively, if the maximum interaction intensity at the level i is high, we would trust more the interaction score of this level. Therefore the M features provide evidence of importance when we assign gating weights to combine the interaction scores of every abstraction level. Then, the M values

are normalized through a softmax gate as follows.

$$\beta_i = \frac{exp(\alpha_i M^{(i)})}{exp(\sum_{j=0}^{L} \alpha_j M^{(j)})} \quad (25)$$

where α_i are learnable parameters, $M^{(i)}$ are the M values for each convolution layer i, and L is the total number of convolution layers.

The interaction scores S_i are then weighted and concatenated as $[\beta_0 S_0, .., \beta_L S_L]$ to be fed into a MLP aggregator to obtain an overall relevance score:

$$S = f(W[\beta_0 S_0, .., \beta_L S_L] + b) \quad (26)$$

The experimental results of multi-level matching models are presented in Table 3, where *Rep_xL_Sy* represents the representation-based model with x convolutional layers, and Sy scores involving in matching. S0+S1 means aggregating S0 and S1 scores. $Inter - Sx$ is the interaction-based model with matching score Sx of level x participating in matching.

Table 3: Results of Multi-level Matching Models[1]

Model	MAP	NDCG@1	NDCG@3	NDCG@10	NDCG@20
BM25	0.0879	0.1178	0.1359	0.1356	0.1394
Rep_0L_S0	0.0614*	0.0862	0.0861*	0.0936*	0.0953*
Rep_1L_S0+S1	0.0401*	0.0857	0.0697*	0.0690*	0.0709*
Rep_2L_S2	0.0121	0.0019*	0.0024	0.0042	0.0053
Rep_2L_S0+S1+S2	0.0503	0.0899*	0.0882	0.0875	0.0857
Rep_3L_S3	0.0145*	0.0158*	0.0143*	0.0143*	0.0149*
Rep_3L_S0+S1+S2	0.0386*	0.0452*	0.0609*	0.0658*	0.0660*
Rep_3L_S0+S1+S2+S3	0.0686*	0.0837*	0.0971*	0.1080*	0.1092*
Inter-S0	0.0546*	0.1218	0.1020*	0.0991*	0.0938*
Inter-S1	0.0789*	0.1218	0.1254	0.1283	0.1272*
Inter-S2	0.0884	0.1485*	0.1423	0.1411	0.1389
Inter-S0+S1+S2	**0.0928**	**0.1610***	**0.1483**	**0.1431**	**0.1424**

From Table 3, we can observe that for both representation- and interaction-based models, the ones employing multi-level matching scores outperform the ones employing only the last level matching score. This result indicates that multi-level matching signals are important to ad hoc tasks. It is also worth noting that with multi-level matching mechanism, the interaction-based model continues to perform better than representation-based models. This result is consistent with the models without multi-level matching presented in the previous section, and further confirms that it is preferable to employ interaction-based model in ad hoc search tasks. In fact, the interaction-based model with multi-level matching $Inter - S0 + S1 + S2$ can even outperform the BM25 baseline.

To compare the multi-level matching representation-based model with interaction-based models, we also plot the learning curve of multi-matching representation- and interaction-based models in Fig 6.

From Fig 6, we can observe that in the case of multi-level matching, interaction-based model still outperforms representation-based model, which indicates the difficulty of training good representations in representation-based models, even when multiple matchings are allowed. In fact, although multiple matching scores allow us to match a query and a document at different levels of abstraction, they are still based on global representations of the document.

[1]* means statistically significant difference with BM25

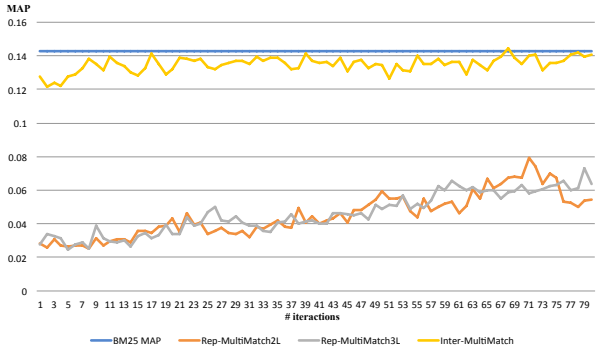


Figure 6: Validation Curve on MAP

These latter continue to have difficulties to capture the important matching elements for different specific queries.

To better understand the usefulness of multi-level matching in representation-based models, we compare the representation-based multi-matching model with some base models on some representative queries in Table 4, where Rep-3L-MultiMatch is the model with matching scores of every level $(S_0 + S_1 + S_2 + S_3)$.

Table 4: nDCG@10 of Representative Queries for Rep Models

Topic_num	Query	Rep-3L_S0	Rep-3L_S3	Rep-3L-MultiMatch
73	Neil Young	0.0600	0.0221	**0.0882**
130	fact on Uranus	0.2976	0.0948	**0.5320**
57	ct jobs	0.0112	0.0226	**0.1121**
77	bobcat	0.0114	0.0406	**0.1862**

From Table 4 we can observe that for lexical queries that ask for exact or near-exact match, such as "Neil Young" (a musician) and "fact on Uranus", the model with only term-level $S0$ matching outperforms the model with only high level score $S3$. The latter may expand too much the semantic of the query therefore results in poor performance. For example, some of the documents retrieved by Multi-Match-3L_S3 for the query "Neil Young" contains other people named Neil which are irrelevant to this query. However, for queries requiring a conceptual match, such as "ct jobs" and "bobcat", the model with high level score $S3$ outperforms the model with low-level score $S0$, since it can learn high level concept representations and perform matching at this level. For example, among the retrieved documents for the query "bobcat", there are desired documents containing information about bobcat company and bobcat brand tractors. In this case the query has been correctly generalized at conceptual level. By combining the matching scores of every level, the Multi-Match-Rep model could outperform both the model with $S0$ and $S3$ scores, which demonstrates the power of multi-level matching.

We also analyze some representative queries for interaction-based models and compare the performance of the interaction-based multi-level matching model with some base models.

For queries which ask for an exact or near-exact match, such as "President of the United States" and "Pocono" (a mountain), the

Table 5: nDCG@10 of Representative Queries for Inter Models

Topic_num	Query	Inter-S0	Inter-S2	Inter-MultiMatch
54	President of the United States	0.1370	0.0367	**0.1429**
153	Pocono	0.0609	0.0595	**0.0658**
145	vines for shade	0.2225	0.3333	**0.4201**
155	last supper painting	0.2240	0.3479	**0.3938**

model with only term-level S_0 matching outperforms the model with high-level score S_2. The latter model may expand too much the semantic of the query, which performs poorly. For example among the documents retrieved for the query "President of the United States" by $Inter - S2$, a number of documents are about presidents of other countries than the United States which are irrelevant for this query. This is an example of over generalization. For conceptual queries such as "vines for shade", "last supper painting", the model with high-level score S_2 outperforms the model with only term-level score S_0, since it captures the high-level interactions required for these queries. For example, the documents retrieved for "last supper painting" by $Inter - S2$ include the desired documents about "description of the last supper painting", "significance of last supper painting in Catholicism". In this case, the query has been correctly generalized at conceptual level. In both above cases, by dynamically combining the matching scores of the 3 levels of abstraction by a gating mechanism, our Multi-Match-Inter model can outperform each of the single-level base models. These examples show the ability of our model to use the appropriate level(s) of matching depending on the query.

6 CONCLUSION AND FUTURE WORK

Representation-based and interaction-based IR models have been proposed in several previous studies, but they have been trained and evaluated on different datasets, which makes it difficult to compare their performance on a fair basis. In this paper, we build convolutional models on both schemas, train them and evalutate their performance on the same training and testing collection. Our first goal is to compare the two approaches as fairly as possible. The experimental results clearly show that interaction-based models always outperform representation-based models. The promising performance of interaction-based model is partly attributed to their capacity to learn the local interaction patterns rather than learning global representations in representation-based model. Our observation is not new, and it is consistent with the previous studies. However, our contribution is to compare the models on a fair basis. This allows us to draw a clear conclusion that interaction-based models are more appropriate than representation-based models.

Moreover, to achieve query-dependent matching, we integrate multi-level matching mechanism into both representation- and interaction-based models and evaluate their performance. The experimental results show that in both schemas, multi-level matching models could outperform models with single-level matching. This result shows the usefulness to design models that can cope with low-level lexical matching as well as high-level semantic matching. In addition. we observe that with multi-level matching mechanism, representation-based model still performs worse than interaction-based model, which further confirms that it is more preferable to employ interaction-based models for ad hoc search.

In this study, we have imposed several constraints in our implementation and experiments in order to compare models as fairly as possible. We deliberately left several useful options such as combining the two schemas, combining with a traditional ranking model (e.g. BM25) in a learning-to-rank framework, and so on. These constraints are also the limitations of this study: we have not tested all the possible options in our comparison. In our future work, we will extend the comparison to models that use more options.

REFERENCES

[1] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of ACM SIGIR '17, Shinjuku, Tokyo, Japan, August 7-11, 2017.* 65–74.

[2] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013.* 6645–6649.

[3] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of ACM CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* 55–64.

[4] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada.* 2042–2050.

[5] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013.* 2333–2338.

[6] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* 1106–1114.

[8] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proceedings of WWW 2017, Perth, Australia, April 3-7, 2017.* 1291–1299.

[9] Gia-Hung Nguyen, Laure Soulier, Lynda Tamine, and Nathalie Bricon-Souf. 2017. DSRIM: A Deep Neural Information Retrieval Model Enhanced by a Knowledge Resource Driven Representation of Documents. In *Proceedings of ACM ICTIR 2017, Amsterdam, The Netherlands, October 1-4, 2017.* 19–26.

[10] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2016. A Study of MatchPyramid Models on Ad-hoc Retrieval. *CoRR* abs/1606.04648 (2016).

[11] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of ACM SIGIR '15 Santiago, Chile, August 9-13, 2015.* 373–382.

[12] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume.* 373–374.

[13] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-Entity Duet Representations for Document Ranking. In *Proceedings of ACM SIGIR '17, Shinjuku, Tokyo, Japan, August 7-11, 2017.* 763–772.

[14] Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *Proceedings of ACM CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* 287–296.

[15] Wenpeng Yin and Hinrich Schütze. 2015. MultiGranCNN: An Architecture for General Matching of Text Chunks on Multiple Levels of Granularity. In *Proceedings of ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers.* 63–73.

[16] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. A support vector method for optimizing average precision. In *Proceedings of ACM SIGIR '07, Amsterdam, The Netherlands, July 23-27, 2007.* 271–278.

[17] Xiaoqiang Zhou, Baotian Hu, Jiaxin Lin, Yang Xiang, and Xiaolong Wang. 2015. ICRC-HIT: A Deep Learning based Comment Sequence Labeling System for Answer Selection Challenge. In *Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, Colorado, USA, June 4-5, 2015.* 210–214.