Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval

Zhenghao Liu[1] **Chenyan Xiong**[2] **Maosong Sun**[1] * **Zhiyuan Liu**[1]
[1]State Key Laboratory of Intelligent Technology and Systems
Beijing National Research Center for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Language Technologies Institute, Carnegie Mellon University

Abstract

This paper presents the Entity-Duet Neural Ranking Model (EDRM), which introduces knowledge graphs to neural search systems. EDRM represents queries and documents by their words and entity annotations. The semantics from knowledge graphs are integrated in the distributed representations of their entities, while the ranking is conducted by interaction-based neural ranking networks. The two components are learned end-to-end, making EDRM a natural combination of entity-oriented search and neural information retrieval. Our experiments on a commercial search log demonstrate the effectiveness of EDRM. Our analyses reveal that knowledge graph semantics significantly improve the generalization ability of neural ranking models.

1 Introduction

The emergence of large scale knowledge graphs has motivated the development of *entity-oriented search*, which utilizes knowledge graphs to improve search engines. The recent progresses in entity-oriented search include better text representations with entity annotations (Xiong et al., 2016; Raviv et al., 2016), richer ranking features (Dalton et al., 2014), entity-based connections between query and documents (Liu and Fang, 2015; Xiong and Callan, 2015), and soft-match query and documents through knowledge graph relations or embeddings (Xiong et al., 2017c; Ensan and Bagheri, 2017). These approaches bring in entities and semantics from knowledge graphs and have greatly improved the effectiveness of feature-based search systems.

*Corresponding author: M. Sun (sms@tsinghua.edu.cn)

Another frontier of information retrieval is the development of neural ranking models (*neural-IR*). Deep learning techniques have been used to learn distributed representations of queries and documents that capture their relevance relations (*representation-based*) (Shen et al., 2014), or to model the query-document relevancy directly from their word-level interactions (*interaction-based*) (Guo et al., 2016a; Xiong et al., 2017b; Dai et al., 2018). Neural-IR approaches, especially the *interaction-based* ones, have greatly improved the ranking accuracy when large scale training data are available (Dai et al., 2018).

Entity-oriented search and neural-IR push the boundary of search engines from two different aspects. Entity-oriented search incorporates human knowledge from entities and knowledge graph semantics. It has shown promising results on feature-based ranking systems. On the other hand, neural-IR leverages distributed representations and neural networks to learn more sophisticated ranking models form large-scale training data. However, it remains unclear how these two approaches interact with each other and whether the entity-oriented search has the same advantage in neural-IR methods as in feature-based systems.

This paper explores the role of entities and semantics in neural-IR. We present an Entity-Duet Neural Ranking Model (EDRM) that incorporates entities in interaction-based neural ranking models. EDRM first learns the distributed representations of entities using their semantics from knowledge graphs: descriptions and types. Then it follows a recent state-of-the-art entity-oriented search framework, the word-entity duet (Xiong et al., 2017a), and matches documents to queries with both bag-of-words and bag-of-entities. Instead of manual features, EDRM uses interaction-based neural models (Dai et al., 2018) to match query and documents with word-entity duet rep-

resentations. As a result, EDRM combines entity-oriented search and the interaction based neural-IR; it brings the knowledge graph semantics to neural-IR and enhances entity-oriented search with neural networks.

One advantage of being neural is that EDRM can be learned end-to-end. Given a large amount of user feedback from a commercial search log, the integration of knowledge graph semantics to neural ranker, is learned jointly with the modeling of query-document relevance in EDRM. It provides a convenient data-driven way to leverage external semantics in neural-IR.

Our experiments on a Sogou query log and CN-DBpedia demonstrate the effectiveness of entities and semantics in neural models. EDRM significantly outperforms the word-interaction-based neural ranking model, K-NRM (Xiong et al., 2017a), confirming the advantage of entities in enriching word-based ranking. The comparison with Conv-KNRM (Dai et al., 2018), the recent state-of-the-art neural ranker that models phrase level interactions, provides a more interesting observation: Conv-KNRM predicts user clicks reasonably well, but integrating knowledge graphs using EDRM significantly improves the neural model's generalization ability on more difficult scenarios.

Our analyses further revealed the source of EDRM's generalization ability: the knowledge graph semantics. If only treating entities as ids and ignoring their semantics from the knowledge graph, the entity annotations are only a cleaner version of phrases. In neural-IR systems, the embeddings and convolutional neural networks have already done a decent job in modeling phrase-level matches. However, the knowledge graph semantics brought by EDRM can not yet be captured solely by neural networks; incorporating those human knowledge greatly improves the generalization ability of neural ranking systems.

2 Related Work

Current neural ranking models can be categorized into two groups: representation based and interaction based (Guo et al., 2016b). The earlier works mainly focus on representation based models. They learn good representations and match them in the learned representation space of query and documents. DSSM (Huang et al., 2013) and its convolutional version CDSSM (Shen et al., 2014) get representations by hashing letter-tri-grams to a low dimension vector. A more recent work uses pseudo-labeling as a weak supervised signal to train the representation based ranking model (Dehghani et al., 2017).

The interaction based models learn word-level interaction patterns from query-document pairs. ARC-II (Hu et al., 2014) and MatchPyramind (Pang et al., 2016) utilize Convolutional Neural Network (CNN) to capture complicated patterns from word-level interactions. The Deep Relevance Matching Model (DRMM) (Guo et al., 2016b) uses pyramid pooling (histogram) to summarize the word-level similarities into ranking models. K-NRM and Conv-KNRM use kernels to summarize word-level interactions with word embeddings and provide soft match signals for learning to rank. There are also some works establishing position-dependent interactions for ranking models (Pang et al., 2017; Hui et al., 2017). Interaction based models and representation based models can also be combined for further improvements (Mitra et al., 2017).

Recently, large scale knowledge graphs such as DBpedia (Auer et al., 2007), Yago (Suchanek et al., 2007) and Freebase (Bollacker et al., 2008) have emerged. Knowledge graphs contain human knowledge about real-word entities and become an opportunity for search system to better understand queries and documents. There are many works focusing on exploring their potential for ad-hoc retrieval. They utilize knowledge as a kind of pseudo relevance feedback corpus (Cao et al., 2008) or weight words to better represent query according to well-formed entity descriptions. Entity query feature expansion (Dietz and Verga, 2014) uses related entity attributes as ranking features.

Another way to utilize knowledge graphs in information retrieval is to build the additional connections from query to documents through related entities. Latent Entity Space (LES) builds an unsupervised model using latent entities' descriptions (Liu and Fang, 2015). EsdRank uses related entities as a latent space, and performs learning to rank with various information retrieval features (Xiong and Callan, 2015). AttR-Duet develops a four-way interaction to involve cross matches between entity and word representations to catch more semantic relevance patterns (Xiong et al., 2017a).

There are many other attempts to integrate

knowledge graphs in neural models in related tasks (Miller et al., 2016; Gupta et al., 2017; Ghazvininejad et al., 2018). Our work shares a similar spirit and focuses on exploring the effectiveness of knowledge graph semantics in neural-IR.

3 Entity-Duet Neural Ranking Model

This section first describes the standard architecture in current interaction based neural ranking models. Then it presents our Entity-Duet Neural Ranking Model, including the semantic entity representation which integrates the knowledge graph semantics, and then the entity-duet ranking framework. The overall architecture of EDRM is shown in Figure 1.

3.1 Interaction based Ranking Models

Given a query q and a document d, interaction based models first build the word-level translation matrix between q and d (Berger and Lafferty, 1999). The translation matrix describes word pairs similarities using word correlations, which are captured by word embedding similarities in interaction based models.

Typically, interaction based ranking models first map each word t in q and d to an L-dimensional embedding $\vec{v}_t$ with an embedding layer Emb_w:

$$\vec{v}_t = \text{Emb}_w(t). \tag{1}$$

It then constructs the interaction matrix M based on query and document embeddings. Each element M^{ij} in the matrix, compares the ith word in q and the jth word in d, e.g. using the cosine similarity of word embeddings:

$$M^{ij} = \cos(\vec{v}_{t_i^q}, \vec{v}_{t_j^d}). \tag{2}$$

With the translation matrix describing the term level matches between query and documents, the next step is to calculate the final ranking score from the matrix. Many approaches have been developed in interaction base neural ranking models, but in general, that would include a feature extractor $\phi()$ on M and then one or several ranking layers to combine the features to the ranking score.

3.2 Semantic Entity Representation

EDRM incorporates the semantic information about an entity from the knowledge graphs into its representation. The representation includes three embeddings: entity embedding, description embedding, and type embedding, all in L dimension and are combined to generate the semantic representation of the entity.

Entity Embedding uses an L-dimensional embedding layer Emb_e to get an entity embedding $\vec{v}_e^{\text{emb}}$ for e:

$$\vec{v}_e^{\text{emb}} = \text{Emb}_e(e). \tag{3}$$

Description Embedding encodes an entity description which contains m words and explains the entity. EDRM first employs the word embedding layer Emb_w to embed the description word w to $\vec{v}_w$. Then it combines all embeddings in text to an embedding matrix $\vec{V}_w$. Next, it leverages convolutional filters to slide over the text and compose the h length n-gram as $\vec{g}_e^j$:

$$\vec{g}_e^j = \text{ReLu}(W_{\text{CNN}} \cdot \vec{V}_w^{j:j+h} + \vec{b}_{\text{CNN}}), \tag{4}$$

where W_{CNN} and $\vec{b}_{\text{CNN}}$ are two parameters of the covolutional filter.

Then we use max pooling after the convolution layer to generate the description embedding $\vec{v}_e^{\text{des}}$:

$$\vec{v}_e^{\text{des}} = \max(\vec{g}_e^1, ..., \vec{g}_e^j, ..., \vec{g}_e^m). \tag{5}$$

Type Embedding encodes the categories of entities. Each entity e has n kinds of types $F_e = \{f_1, ..., f_j, ..., f_n\}$. EDRM first gets the f_j embedding $\vec{v}_{f_j}$ through the type embedding layer Emb_{tp}:

$$\vec{v}_{f_j}^{\text{emb}} = \text{Emb}_{\text{tp}}(e). \tag{6}$$

Then EDRM utilizes an attention mechanism to combine entity types to the type embedding $\vec{v}_e^{\text{type}}$:

$$\vec{v}_e^{\text{type}} = \sum_j^n a_j \vec{v}_{f_j}, \tag{7}$$

where a_j is the attention score, calculated as:

$$a_j = \frac{\exp(P_j)}{\sum_l^n \exp(P_l)}, \tag{8}$$

$$P_j = (\sum_i W_{\text{bow}} \vec{v}_{t_i}) \cdot \vec{v}_{f_j}. \tag{9}$$

P_j is the dot product of the query or document representation and type embedding f_j. We leverage bag-of-words for query or document encoding. W_{bow} is a parameter matrix.

Combination. The three embeddings are combined by an linear layer to generate the semantic representation of the entity:

$$\vec{v}_e^{\text{sem}} = \vec{v}_e^{\text{emb}} + W_e(\vec{v}_e^{\text{des}} \oplus \vec{v}_e^{\text{type}})^T + \vec{b}_e. \tag{10}$$

W_e is an $L \times 2L$ matrix and $\vec{b}_e$ is an L-dimensional vector.

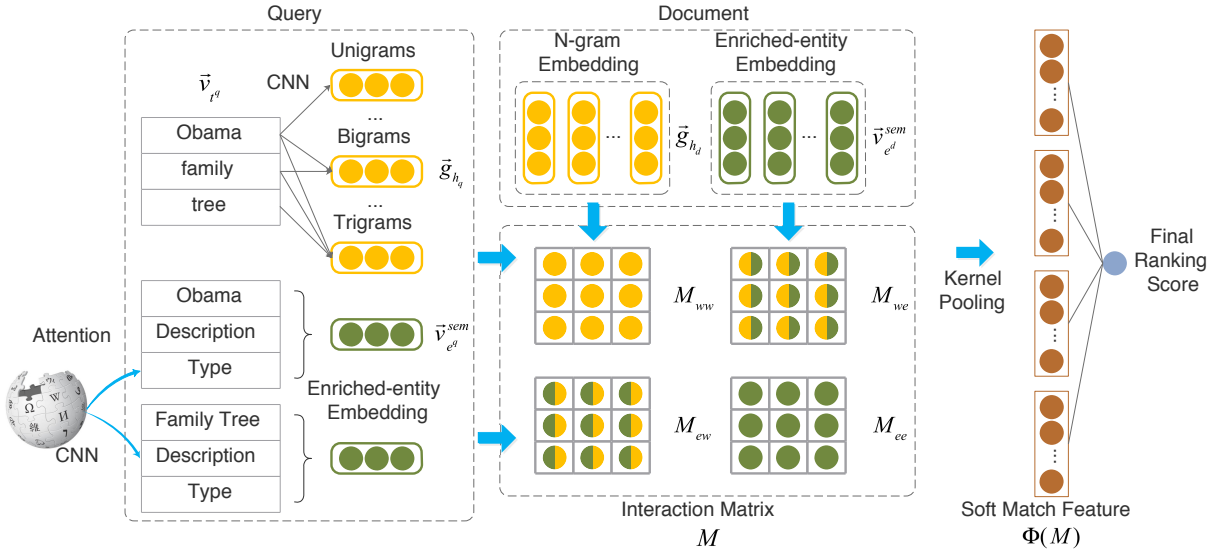


Figure 1: The architecture of EDRM.

3.3 Neural Entity-Duet Framework

Word-entity duet (Xiong et al., 2017a) is a recently developed framework in entity-oriented search. It utilizes the duet representation of bag-of-words and bag-of-entities to match q-d with hand crafted features. This work introduces it to neural-IR.

We first construct bag-of-entities q^e and d^e with entity annotation as well as bag-of-words q^w and d^w for q and d. The duet utilizes a four-way interaction: query words to document words (q^w-d^w), query words to documents entities (q^w-d^e), query entities to document words (q^e-d^w) and query entities to document entities (q^e-d^e).

Instead of features, EDRM uses a translation layer that calculates similarity between a pair of query-document terms: ($\vec{v}^i_{w^q}$ or $\vec{v}^i_{e^q}$) and ($\vec{v}^j_{w^d}$ or $\vec{v}^j_{e^d}$). It constructs the interaction matrix $M = \{M_{ww}, M_{we}, M_{ew}, M_{ee}\}$. And $M_{ww}, M_{we}, M_{ew}, M_{ee}$ denote interactions of q^w-d^w, q^w-d^e, q^e-d^w, q^e-d^e respectively. And elements in them are the cosine similarities of corresponding terms:

$$M^{ij}_{ww} = \cos(\vec{v}^i_{w^q}, \vec{v}^j_{w^d}); M^{ij}_{ee} = \cos(\vec{v}^i_{e^q}, \vec{v}^j_{e^d})$$
$$M^{ij}_{ew} = \cos(\vec{v}^i_{e^q}, \vec{v}^j_{w^d}); M^{ij}_{we} = \cos(\vec{v}^i_{w^q}, \vec{v}^j_{e^d}). \quad (11)$$

The final ranking feature $\Phi(\mathcal{M})$ is a concatenation ($\oplus$) of four cross matches ($\phi(M)$):

$$\Phi(\mathcal{M}) = \phi(M_{ww}) \oplus \phi(M_{we}) \oplus \phi(M_{ew}) \oplus \phi(M_{ee}), \quad (12)$$

where the ϕ can be any function used in interaction based neural ranking models.

The entity-duet presents an effective way to cross match query and document in entity and word spaces. In EDRM, it introduces the knowledge graph semantics representations into neural-IR models.

4 Integration with Kernel based Neural Ranking Models

The duet translation matrices provided by EDRM can be plugged into any standard interaction based neural ranking models. This section expounds special cases where it is integrated with K-NRM (Xiong et al., 2017b) and Conv-KNRM (Dai et al., 2018), two recent state-of-the-arts.

K-NRM uses K Gaussian kernels to extract the matching feature $\phi(M)$ from the translation matrix M. Each kernel K_k summarizes the translation scores as soft-TF counts, generating a K-dimensional feature vector $\phi(M) = \{K_1(M), ..., K_K(M)\}$:

$$K_k(M) = \sum_j \exp(-\frac{M^{ij} - \mu_k}{2\delta_k^2}). \quad (13)$$

μ_k and δ_k are the mean and width for the kth kernel. Conv-KNRM extend K-NRM incorporating h-gram compositions $\vec{g}^i_h$ from text embedding $\vec{V}_T$ using CNN:

$$\vec{g}^i_h = \text{relu}(W_h \cdot \vec{V}^{i:i+h}_T + \vec{v}_h). \quad (14)$$

Then a translation matrix M_{h_q,h_d} is constructed. Its elements are the similarity scores of h-gram

pairs between query and document:

$$M_{h_q,h_d} = \cos(\vec{g}^i_{h_q}, \vec{g}^j_{h_d}). \quad (15)$$

We also extend word n-gram cross matches to word entity duet matches:

$$\Phi(\mathcal{M}) = \phi(M_{1,1}) \oplus ... \oplus \phi(M_{h_q,h_d}) \oplus ... \oplus \phi(M_{ee}). \quad (16)$$

Each ranking feature $\phi(M_{h_q,h_d})$ contains three parts: query h_q-gram and document h_d-gram match feature ($\phi(M_{ww^{h_q,h_d}})$), query entity and document h_d-gram match feature ($\phi(M_{ew^{1,h_d}})$), and query h_q-gram and document entity match feature ($\phi(M_{ww^{h_q,1}})$):

$$\phi(M_{h_q,h_d}) = \phi(M_{ww^{h_q,h_d}}) \oplus \phi(M_{ew^{1,h_d}}) \oplus \phi(M_{we^{h_q,1}}). \quad (17)$$

We then use learning to rank to combine ranking feature $\Phi(\mathcal{M})$ to produce the final ranking score:

$$f(q,d) = \tanh(\omega_r^T \Phi(\mathcal{M}) + b_r). \quad (18)$$

ω_r and b_r are the ranking parameters. tanh is the activation function.

We use standard pairwise loss to train the model:

$$l = \sum_q \sum_{d^+,d^- \in D_q^{+,-}} \max(0, 1 - f(q,d^+) + f(q,d^-)), \quad (19)$$

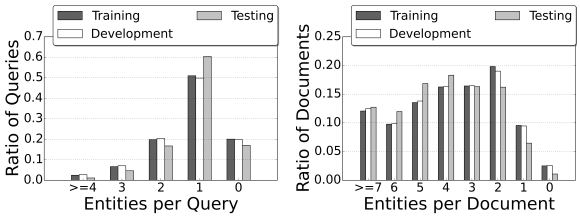
where the d^+ is a document ranks higher than d^-.

With sufficient training data, the whole model is optimized end-to-end with back-propagation. During the process, the integration of the knowledge graph semantics, entity embedding, description embeddings, type embeddings, and matching with entities-are learned jointly with the ranking neural network.

5 Experimental Methodology

This section describes the dataset, evaluation metrics, knowledge graph, baselines, and implementation details of our experiments.

Dataset. Our experiments use a query log from Sogou.com, a major Chinese searching engine (Luo et al., 2017). The exact same dataset and training-testing splits in the previous research (Xiong et al., 2017b; Dai et al., 2018) are used. They defined the ad-hoc ranking task in this dataset as re-ranking the candidate documents provided by the search engine. All Chinese texts are segmented by ICTCLAS (Zhang et al., 2003), after that they are treated the same as English.



(a) Statistic of queries (b) Statistic of documents

Figure 2: Query and document distributions. Queries and documents are grouped by the number of entities.

Prior research leverages clicks to model user behaviors and infer reliable relevance signals using click models (Chuklin et al., 2015). DCTR and TACM are two click models: DCTR calculates the relevance scores of a query-document pair based on their click through rates (CTR); TACM (Wang et al., 2013) is a more sophisticated model that uses both clicks and dwell times. Following previous research (Xiong et al., 2017b), both DCTR and TACM are used to infer labels. DCTR inferred relevance labels are used in training. Three testing scenarios are used: Testing-SAME, Testing-DIFF and Testing-RAW.

Testing-SAME uses DCTR labels, the same as in training. Testing-DIFF evaluates models performance based on TACM inferred relevance labels. Testing-RAW evaluates ranking models through user clicks, which tests ranking performance for the most satisfying document. Testing-DIFF and Testing-RAW are harder scenarios that challenge the generalization ability of all models, because their training labels and testing labels are generated differently (Xiong et al., 2017b).

Evaluation Metrics. NDCG@1 and NDCG@10 are used in Testing-SAME and Testing-DIFF. MRR is used for Testing-Raw. Statistic significances are tested by permutation test with P<0.05. All are the same as in previous research (Xiong et al., 2017b).

Knowledge Graph. We use CN-DBpedia (Xu et al., 2017), a large scale Chinese knowledge graph based on Baidu Baike, Hudong Baike, and Chinese Wikipedia. CN-DBpedia contains 10,341,196 entities and 88,454,264 relations. The query and document entities are annotated by CMNS, the commonness (popularity) based entity linker (Hasibi et al., 2017). CN-DBpedia and CMNS provide good coverage on our queries and

Table 1: Ranking accuracy of EDRM-KNRM, EDRM-CKNRM and baseline methods. Relative performances compared with K-NRM are in percentages. †, ‡, §, ¶, ∗ indicate statistically significant improvements over DRMM†, CDSSM‡, MP§, K-NRM¶ and Conv-KNRM∗ respectively.

	Testing-SAME				Testing-DIFF				Testing-RAW	
Method	**NDCG@1**		**NDCG@10**		**NDCG@1**		**NDCG@10**		**MRR**	
BM25	0.1422	−46.24%	0.2868	−31.67%	0.1631	−45.63%	0.3254	−23.04%	0.2280	−33.86%
RankSVM	0.1457	−44.91%	0.3087	−26.45%	0.1700	−43.33%	0.3519	−16.77%	0.2241	−34.99%
Coor-Ascent	0.1594	−39.74%	0.3547	−15.49%	0.2089	−30.37%	0.3775	−10.71%	0.2415	−29.94%
DRMM	0.1367	−48.34%	0.3134	−25.34%	0.2126‡	−29.14%	0.3592§	−15.05%	0.2335	−32.26%
CDSSM	0.1441	−45.53%	0.3329	−20.69%	0.1834	−38.86%	0.3534	−16.41%	0.2310	−33.00%
MP	0.2184†‡	−17.44%	0.3792†‡	−9.67%	0.1969	−34.37%	0.3450	−18.40%	0.2404	−30.27%
K-NRM	0.2645	–	0.4197	–	0.3000	–	0.4228	–	0.3447	–
Conv-KNRM	0.3357†‡§¶	+26.90%	0.4810†‡§¶	+14.59%	0.3384†‡§¶	+12.81%	0.4318†‡§	+2.14%	0.3582†‡§	+3.91%
EDRM-KNRM	0.3096†‡§¶	+17.04%	0.4547†‡§¶	+8.32%	0.3327†‡§¶	+10.92%	0.4341†‡§¶	+2.68%	0.3616†‡§¶	+4.90%
EDRM-CKNRM	**0.3397**†‡§¶	+28.42%	**0.4821**†‡§¶	+14.86%	**0.3708**†‡§¶∗	+23.60%	**0.4513**†‡§¶∗	+6.74%	**0.3892**†‡§¶∗	+12.90%

documents. As shown in Figure 2, the majority of queries have at least one entity annotation; the average number of entity annotated per document title is about four.

Baselines. The baselines include feature-based ranking models and neural ranking models. Most of the baselines are borrowed from previous research (Xiong et al., 2017b; Dai et al., 2018).

Feature-based baselines include two learning to rank systems, RankSVM (Joachims, 2002) and coordinate ascent (Coor-Accent) (Metzler and Croft, 2006). The standard word-based unsupervised retrieval model, BM25, is also compared.

Neural baselines include CDSSM (Shen et al., 2014), MatchPyramid (MP) (Pang et al., 2016), DRMM (Grauman and Darrell, 2005), K-NRM (Xiong et al., 2017b) and Conv-KNRM (Dai et al., 2018). CDSSM is representation based. It uses CNN to build query and document representations on word letter-tri-grams (or Chinese characters). MP and DRMM are both interaction based models. They use CNNs or histogram pooling to extract features from embedding based translation matrix.

Our main baselines are K-NRM and Conv-KNRM, the recent state-of-the-art neural models on the Sogou-Log dataset. The goal of our experiments is to explore the effectiveness of knowledge graphs in these state-of-the-art interaction based neural models.

Implementation Details. The dimension of word embedding, entity embedding and type embedding are 300. Vocabulary size of entities and words are 44,930 and 165,877. Conv-KNRM uses one layer CNN with 128 filter size for the n-gram composition. Entity description encoder is a one layer CNN with 128 and 300 filter size for Conv-KNRM and K-NRM respectively.

All models are implemented with PyTorch. Adam is utilized to optimize all parameters with learning rate $= 0.001$, $\epsilon = 1e-5$ and early stopping with the practice of 5 epochs.

There are two versions of EDRM: EDRM-KNRM and EDRM-CKNRM, integrating with K-NRM and Conv-KNRM respectively. The first one (K-NRM) enriches the word based neural ranking model with entities and knowledge graph semantics; the second one (Conv-KNRM) enriches the n-gram based neural ranking model.

6 Evaluation Results

Four experiments are conducted to study the effectiveness of EDRM: the overall performance, the contributions of matching kernels, the ablation study, and the influence of entities in different scenarios. We also do case studies to show effect of EDRM on document ranking.

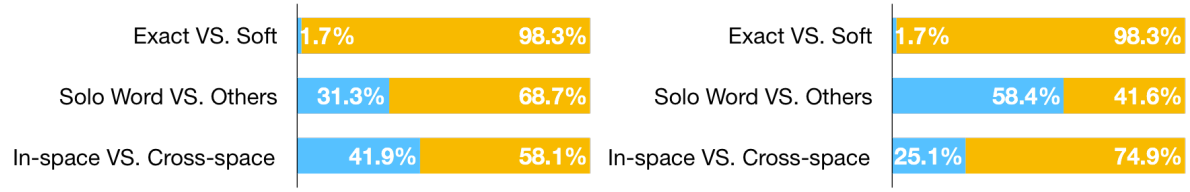
6.1 Ranking Accuracy

The ranking accuracies of the ranking methods are shown in Table 1. K-NRM and Conv-KNRM outperform other baselines in all testing scenarios by large margins as shown in previous research.

EDRM-KNRM out performs K-NRM by over 10% improvement in Testing-SAME and Testing-DIFF. EDRM-CKNRM has almost same performance on Testing-SAME with Conv-KNRM. A possible reason is that, entity annotations provide effective phrase matches, but Conv-KNRM is also able to learn phrases matches automatically from data. However, EDRM-CKNRM has significant improvement on Testing-DIFF and Testing-RAW. Those demonstrate that EDRM has strong ability to overcome domain differences from different labels.

Table 2: Ranking accuracy of adding diverse semantics based on K-NRM and Conv-KNRM. Relative performances compared are in percentages. †, ‡, §, ¶, ∗, ∗∗ indicate statistically significant improvements over K-NRM† (or Conv-KNRM†), +Embed‡, +Type§, +Description¶, +Embed+Type∗ and +Embed+Description∗∗ respectively.

	Testing-SAME				Testing-DIFF				Testing-RAW	
Method	**NDCG@1**		**NDCG@10**		**NDCG@1**		**NDCG@10**		**MRR**	
K-NRM	0.2645	–	0.4197	–	0.3000	–	0.4228	–	0.3447	–
+Embed	0.2743	+3.68%	0.4296	+2.35%	0.3134	+4.48%	0.4306	+1.86%	0.3641†	+5.62%
+Type	0.2709	+2.41%	0.4395†	+4.71%	0.3126	+4.20%	0.4373†	+3.43%	0.3531	+2.43%
+Description	0.2827	+6.86%	0.4364†	+3.97%	0.3181	+6.04%	0.4306	+1.86%	**0.3691**†§∗	+7.06%
+Embed+Type	0.2924†	+10.52%	0.4533†‡§¶	+8.00%	0.3034	+1.13%	0.4297	+1.65%	0.3544	+2.79%
+Embed+Description	0.2891	+9.29%	0.4443†‡	+5.85%	0.3197	+6.57%	0.4304	+1.80%	0.3564	+3.38%
Full Model	**0.3096**†‡§	+17.04%	**0.4547**†‡§¶	+8.32%	**0.3327**†∗	+10.92%	**0.4341**†	+2.68%	0.3616†	+4.90%
Conv-KNRM	0.3357	–	0.4810	–	0.3384	–	0.4318	–	0.3582	–
+Embed	0.3382	+0.74%	**0.4831**	+0.44%	0.3450	+1.94%	0.4413	+2.20%	0.3758†	+4.91%
+Type	0.3370	+0.38%	0.4762	−0.99%	0.3422	+1.12%	0.4423†	+2.42%	0.3798†	+6.02%
+Description	0.3396	+1.15%	0.4807	−0.05%	0.3533	+4.41%	0.4468†	+3.47%	0.3819†	+6.61%
+Embed+Type	**0.3420**	+1.88%	0.4828	+0.39%	0.3546	+4.79%	0.4491†	+4.00%	0.3805†	+6.22%
+Embed+Description	0.3382	+0.73%	0.4805	−0.09%	0.3608	+6.60%	0.4494†	+4.08%	0.3868†	+7.99%
Full Model	0.3397	+1.19%	0.4821	+0.24%	**0.3708**†‡§	+9.57%	**0.4513**†‡	+4.51%	**0.3892**†‡	+8.65%



(a) Kernel weight distribution for EDRM-KNRM.

(b) Kernel weight distribution for EDRM-CKNRM.

Figure 3: Ranking contribution for EDRM. Three scenarios are presented: Exact VS. Soft compares the weights of exact match kernel and others; Solo Word VS. Others shows the proportion of only text based matches; In-space VS. Cross-space compares in-space matches and cross-space matches.

These results show the effectiveness and the generalization ability of EDRM. In the following experiments, we study the source of this generalization ability.

6.2 Contributions of Matching Kernels

This experiment studies the contribution of knowledge graph semantics by investigating the weights learned on the different types of matching kernels.

As shown in Figure 3(a), most of the weight in EDRM-KNRM goes to soft match (Exact VS. Soft); entity related matches play an as important role as word based matches (Solo Word VS. Others); cross-space matches are more important than in-space matches (In-space VS. Cross-space). As shown in Figure 3(b), the percentages of word based matches and cross-space matches are more important in EDRM-CKNRM compared to in EDRM-KNRM.

The contribution of each individual match type in EDRM-CKNRM is shown in Figure 4. The weight of unigram, bigram, trigram, and entity is almost uniformly distributed, indicating the effectiveness of entities and all components are important in EDRM-CKNRM.

6.3 Ablation Study

This experiment studies which part of the knowledge graph semantics leads to the effectiveness and generalization ability of EDRM.

There are three types of embeddings incorporating different aspects of knowledge graph information: entity embedding (Embed), description embedding (Description) and type embedding (Type). This experiment starts with the word-only K-NRM and Conv-KNRM, and adds these three types of embedding individually or two-by-two (Embed+Type and Embed+Description).

The performances of EDRM with different groups of embeddings are shown in Table 2. The description embeddings show the greatest improvement among the three embeddings. Entity

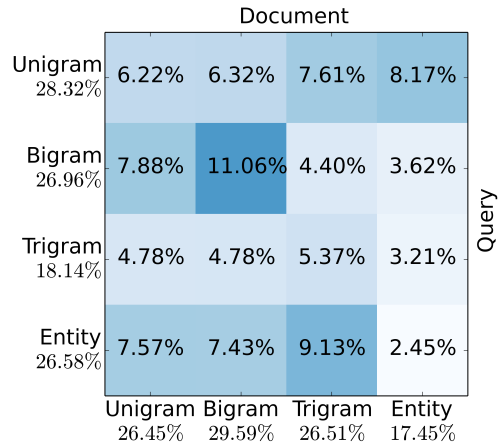


Figure 4: Individual kernel weight for EDRM-CKNRM. X-axis and y-axis denote document and query respectively.

type plays an important role only combined with other embeddings. Entity embedding improves K-NRM while has little effect on Conv-KNRM. This result further confirms that the signal from entity names are captured by the n-gram CNNs in Conv-KNRM. Incorporating all of three embeddings usually gets the best ranking performance.

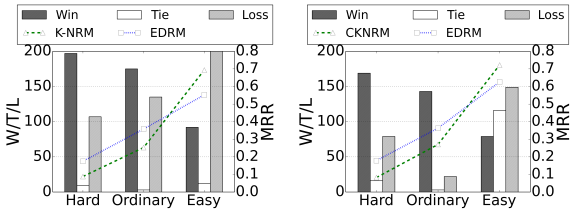
This experiments shows that knowledge graph semantics are crucial to EDRM's effectiveness. Conv-KNRM learns good phrase matches that overlap with the entity embedding signals. However, the knowledge graph semantics (descriptions and types) is hard to be learned just from user clicks.

6.4 Performance on Different Scenarios

This experiment analyzes the influence of knowledge graphs in two different scenarios: multiple difficulty degrees and multiple length degrees.

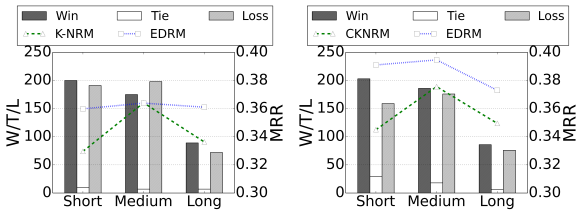
Query Difficulty Experiment studies EDRM's performance on testing queries at different difficulty, partitioned by Conv-KNRM's MRR value: Hard (MRR $<$ 0.167), Ordinary (MRR $\in [0.167, 0.382]$, and Easy (MRR $>$ 0.382). As shown in Figure 5, EDRM performs the best on hard queries.

Query Length Experiment evaluates EDRM's effectiveness on Short (1 words), Medium (2-3 words) and Long (4 or more words) queries. As shown in Figure 6, EDRM has more win cases and achieves the greatest improvement on short queries. Knowledge embeddings are more crucial when limited information is available from the original query text.



(a) K-NRM VS. EDRM (b) Conv-KNRM VS. EDRM

Figure 5: Performance VS. Query Difficulty. The x-axises mark three query difficulty levels. The y-axises are the Win/Tie/Loss (left) and MRR (right) in the corresponding group.



(a) K-NRM VS. EDRM (b) Conv-KNRM VS. EDRM

Figure 6: Performance VS. Query Length. The x-axises mark three query length levels, and y-axises are the Win/Tie/Loss (left) and MRR (right) in the corresponding group.

These two experiments reveal that the effectiveness of EDRM is more observed on harder or shorter queries, whereas the word-based neural models either find it difficult or do not have sufficient information to leverage.

6.5 Case Study

Table 3 provide examples reflecting two possible ways, in which the knowledge graph semantics could help the document ranking.

First, the entity descriptions explain the meaning of entities and connect them through the word space. *Meituxiuxiu web version* and *Meilishuo* are two websites providing image processing and shopping services respectively. Their descriptions provide extra ranking signals to promote the related documents.

Second, entity types establish underlying relevance patterns between query and documents. The underlying patterns can be captured by cross-space matches. For example, the types of the query entity *Crayon Shin-chan* and *GINTAMA* overlaps with the bag-of-words in the relevant documents. They can also be captured by the entity-based matches through their type overlaps,

Table 3: Examples of entity semantics connecting query and title. All the examples are correctly ranked by EDRM-CKNRM. Table 3a shows query-document pairs. Table 3b lists the related entity semantics that include useful information to match the query-document pair. The examples and related semantics are picked by manually examining the ranking changes between different variances of EDRM-CKNRM.

(a) Query and document examples. *Entities* are emphasized.

Query	Document
Meituxiuxiu web version	*Meituxiuxiu web version*: An online picture processing tools
Home page of *Meilishuo*	Home page of *Meilishuo* - Only the correct popular fashion
Master Lu	Master Lu official website: *System optimization*, hardware test, phone evaluation
Crayon Shin-chan: The movie	*Crayon Shin-chan*: The movie online - Anime
GINTAMA	*GINTAMA*: The movie online - Anime - Full HD online watch

(b) Semantics of related entities. The first two rows and last two rows show entity descriptions and entity types respectively.

Entity	Content
Meituxiuxiu web version	Description: Meituxiuxiu is the most popular Chinese image processing software, launched by the Meitu company
Meilishuo	Description: Meilishuo, the largest women's fashion e-commerce platform, dedicates to provide the most popular fashion shopping experience
Crayon Shin-chan, *GINTAMA*	Type: Anime; Cartoon characters; Comic
Master Lu, *System Optimization*	Type: Hardware test; Software; System tool

for example, between the query entity *Master Lu* and the document entity *System Optimization*.

7 Conclusions

This paper presents EDRM, the Entity-Duet Neural Ranking Model that incorporating knowledge graph semantics into neural ranking systems. EDRM inherits entity-oriented search to match query and documents with bag-of-words and bag-of-entities in neural ranking models. The knowledge graph semantics are integrated as distributed representations of entities. The neural model leverages these semantics to help document ranking. Using user clicks from search logs, the whole model—the integration of knowledge graph semantics and the neural ranking networks–is trained end-to-end. It leads to a data-driven combination of entity-oriented search and neural information retrieval.

Our experiments on the Sogou search log and CN-DBpedia demonstrate EDRM's effectiveness and generalization ability over two state-of-the-art neural ranking models. Our further analyses reveal that the generalization ability comes from the integration of knowledge graph semantics. The neural ranking models can effectively model n-gram matches between query and document, which overlaps with part of the ranking signals from entity-based matches: Solely adding the entity names may not improve the ranking accuracy much. However, the knowledge graph semantics, introduced by the description and type embeddings, provide novel ranking signals that greatly improve the generalization ability of neural rankers in difficult scenarios.

This paper preliminarily explores the role of structured semantics in deep learning models. Though mainly fouced on search, we hope our findings shed some lights on a potential path towards more intelligent neural systems and will motivate more explorations in this direction.

Acknowledgments

This work[1] is supported by the Major Project of the National Social Science Foundation of China (No.13&ZD190) as well as the China-Singapore Joint Research Project of the National Natural Science Foundation of China (No. 61661146007) under the umbrella of the NexT Joint Research Center of Tsinghua University and National University of Singapore. Chenyan Xiong is supported by National Science Foundation (NSF) grant IIS-1422676. We thank Sogou for providing access to the search log.

[1] Source codes of this work are available at http://github.com/thunlp/EntityDuetNeuralRanking

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. *DBpedia: A nucleus for a web of open data*. Springer.

Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*. ACM, pages 222–229.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*. ACM, pages 1247–1250.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*. ACM, pages 243–250.

Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7(3):1–115.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018)*. ACM, pages 126–134.

Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2014)*. ACM, pages 365–374.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, pages 65–74.

Laura Dietz and Patrick Verga. 2014. Umass at TREC 2014: Entity query feature expansion using knowledge base links. In *Proceedings of The 23st Text Retrieval Conference (TREC 2014)*. NIST.

Faezeh Ensan and Ebrahim Bagheri. 2017. Document retrieval model through semantic linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM 2017)*. ACM, pages 181–190.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Scott Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*.

Kristen Grauman and Trevor Darrell. 2005. The pyramid match kernel: Discriminative classification with sets of image features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. IEEE, volume 2, pages 1458–1465.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016a. Semantic matching by non-linear word transportation for information retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016)*. ACM, pages 701–710.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W.Bruce Croft. 2016b. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016)*. ACM, pages 55–64.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. pages 2681–2690.

Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2017. Entity linking in queries: Efficiency vs. effectiveness. In *European Conference on Information Retrieval*. Springer, pages 40–53.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS 2014)*. MIT Press, pages 2042–2050.

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 *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (CIKM 2013)*. ACM, pages 2333–2338.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. Pacrr: A position-aware neural ir model for relevance matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. pages 1060–1069.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002)*. ACM, pages 133–142.

Xitong Liu and Hui Fang. 2015. Latent entity space: A novel retrieval approach for entity-bearing queries. *Information Retrieval Journal* 18(6):473–503.

Cheng Luo, Yukun Zheng, Yiqun Liu, Xiaochuan Wang, Jingfang Xu, Min Zhang, and Shaoping Ma. 2017. Sogout-16: A new web corpus to embrace ir research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, pages 1233–1236.

Donald Metzler and W. Bruce Croft. 2006. Linear feature-based models for information retrieval. *Information Retrieval* 10(3):257–274.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. pages 1400–1409.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. ACM, pages 1291–1299.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*. pages 2793–2799.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017)*. ACM, pages 257–266.

Hadas Raviv, Oren Kurland, and David Carmel. 2016. Document retrieval using entity-based language models. In *Proceedings of the 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*. ACM, pages 65–74.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM 2014)*. ACM, pages 101–110.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web (WWW 2007)*. ACM, pages 697–706.

Hongning Wang, ChengXiang Zhai, Anlei Dong, and Yi Chang. 2013. Content-aware click modeling. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW 2013)*. ACM, pages 1365–1376.

Chenyan Xiong and Jamie Callan. 2015. EsdRank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM 2015)*. ACM, pages 951–960.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2016. Bag-of-entities representation for ranking. In *Proceedings of the sixth ACM International Conference on the Theory of Information Retrieval (ICTIR 2016)*. ACM, pages 181–184.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017a. Word-entity duet representations for document ranking. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, pages 763–772.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017b. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, pages 55–64.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017c. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. ACM, pages 1271–1279.

Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. 2017. Cndbpedia: A never-ending chinese knowledge extraction system. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, pages 428–438.

Hua Ping Zhang, Hong Kui Yu, De Yi Xiong, and Qun Liu. 2003. Hhmm-based chinese lexical analyzer ictclas. In *Sighan Workshop on Chinese Language Processing*. pages 758–759.