information

MDPI

Article

Exploring the Importance of Entities in Semantic Ranking

Zhenyang Li [1,2,3], Guangluan Xu [1,2,3], Xiao Liang [1,2], Feng Li [1,2], Lei Wang [1,2] and Daobing Zhang [1,2,*]

[1] Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; lizhenyang16@mails.ucas.ac.cn (Z.L.); gluanxu@mail.ie.ac.cn (G.X.); xliang@mail.ie.ac.cn (X.L.); lifeng@mail.ie.ac.cn (F.L.); visualwang@139.com (L.W.)

[2] Key Laboratory of Network Information System Technology (NIST), Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China

[3] University of Chinese Academy of Sciences, Beijing 100049, China

[*] Correspondence: dbzhang@mail.ie.ac.cn; Tel.: +86-138-1176-3576

Received: 16 November 2018; Accepted: 17 January 2019; Published: 24 January 2019

Abstract: In recent years, entity-based ranking models have led to exciting breakthroughs in the research of information retrieval. Compared with traditional retrieval models, entity-based representation enables a better understanding of queries and documents. However, the existing entity-based models neglect the importance of entities in a document. This paper attempts to explore the effects of the importance of entities in a document. Specifically, the dataset analysis is conducted which verifies the correlation between the importance of entities in a document and document ranking. Then, this paper enhances two entity-based models—toy model and Explicit Semantic Ranking model (ESR)—by considering the importance of entities. In contrast to the existing models, the enhanced models assign the weights of entities according to their importance. Experimental results show that the enhanced toy model and ESR can outperform the two baselines by as much as 4.57% and 2.74% on NDCG@20 respectively, and further experiments reveal that the strength of the enhanced models is more evident on long queries and the queries where ESR fails, confirming the effectiveness of taking the importance of entities into account.

Keywords: information retrieval; entity-based ranking; importance of entities

1. Introduction

Information retrieval (IR), which facilitates the users in search of the required information, has always been a research hotspot. In traditional IR, words are fundamental to the representation of queries and documents [1–3]. The early research (such as vector space model [4] and BM25 [5]) typically built bag-of-words representations for texts, and the relevance between query and document is usually based on handcrafting features. Recently, many types of research have also adopted text embeddings and neural networks, and these neural information retrieval models could extract IR features from training data automatically (such as DSSM [2] and DRMM [3]). However, in the previous IR models which represent texts based on words, the information obtained from the surface forms is insufficient to rank documents. Later, with more abundant semantic information provided by knowledge graphs for document-ranking, entity-based representation has become a direction worth exploring. As shown in Figure 1, in entity-based ranking models [6–8], the queries and documents are represented by entities with the help of entity linking [9,10]. Compared with previous IR models, the entity-based ranking models incorporate external knowledge with a document-ranking task, since entities and their semantics in the knowledge graph enable a better understanding of queries and documents.

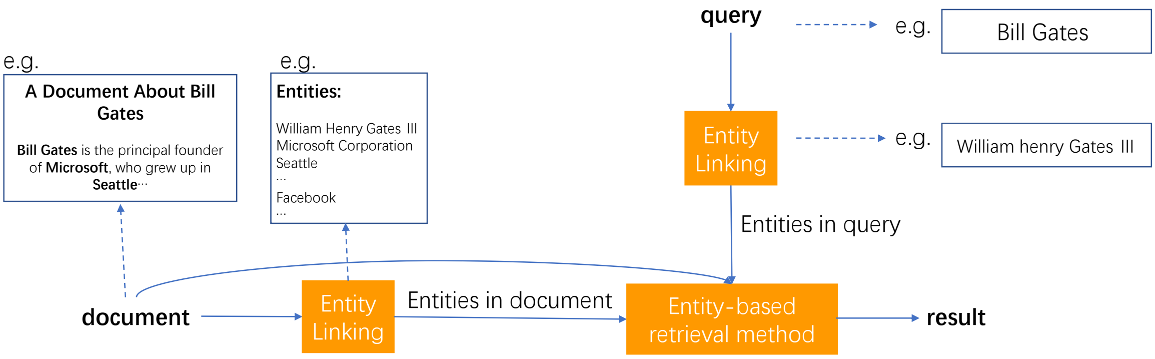


Figure 1. The framework of entity-based model.

Although much research has been carried out on entity-based models, little has investigated the importance of entities in documents. For an entity in a document, its importance depends on how much it contributes to the semantics of this document. For example, in a document about 'Bill Gates', the importance of the entities 'Microsoft' and 'Seattle', which are related to 'Bill Gates', should be higher than the importance of 'Facebook'. In traditional IR models, weights have always been assigned to words according to their importance, and proper term-weighting can greatly improve the accuracy of retrieval systems. Intuitively, for entity-based models, the importance of entities should also be considered. Nevertheless, entities in a document are equally treated in most existing entity-based models. Thus, this paper attempts to explore the effectiveness of the importance of entities in a document.

This paper mainly conducts research into two aspects. Firstly, data analysis is carried out on the ESR dataset—an academic dataset used for entity-based models. TextRank and term frequency (TF)-inverse document frequency (IDF) are applied to measure the importance of entities, respectively. The results of the analysis reveal that the importance of entities in a document is indeed related to document ranking. Secondly, based on the analysis, two entity-based models (toy model and ESR) are optimized by considering the importance of entities. This work assigns high weights to the important entities in documents and low weights to the less important ones. Extensive experiments on the ESR dataset confirm the effectiveness of considering the importance of entities. With TextRank or TF-IDF adopted, both two entity-based models perform better. The enhanced toy model achieves a 4.57% increase in NDCG@20 over toy model. In addition, the enhanced ESR achieves a 2.79% increase in NDCG@20 over ESR, even merely using the high-ranking entities can achieve a 0.88% increase. In addition, the effectiveness of the enhanced ESR models is more obvious on hard and long queries.

In summary, the main contributions of this paper are as follows:

1. The dataset analysis reveals that the importance of entities in a document should be considered in the IR task.
2. Toy model and ESR are enhanced by considering the importance of entities in documents.
3. Extensive experiments reveal that the enhanced models could achieve better performance than the original models on all evaluation metrics, especially on the long queries and the queries where ESR fails.

This paper is structured as follows: Section 2 reviews the previous works related to entity-based IR and entity ranking. Section 3 introduces entity ranking methods and analyzes the ESR dataset. The approaches for using entity ranking in toy model and ESR are described in Section 4. Section 5 presents the experimental settings and experimental results. Finally, conclusions and inspiration for future studies are offered in Section 6.

2. Related Work

2.1. Traditional Information Retrieval Methods

Early works about IR models are mostly based on exact matching of terms between queries and documents [11]. The matching score can be computed by vector space model [4] or more complex methods such as BM25 [5]. These traditional models mainly use bag-of-words representation for both queries and documents, which results in the limitation of retrieving documents that are just syntactically relevant, rather than semantically relevant [12].

Recent years, while deep learning and word embedding [13] have been successfully applied in IR, neural ranking models have been gaining much concern. By matching texts to embeddings, the semantic matches between queries and documents can be computed in the representation space [14]. Current neural IR models can be categorized into two groups: representation-based and interaction-based models [1]. The representation-based models first build representations for the query and documents and then conduct matching in the embedding space. DSSM [2] and CDSSM [15] get low-dimensional vectors representations using deep neural networks and handle the out-of-vocabulary problem (OOV) by hashing letter-tri-grams to a low dimension vector.

The interaction-based models first get the word-level interactions from query-document pairs and then use deep neural networks to learn patterns for matching. For example, ARC-II [16] and MatchPyramid [17] employ a CNN to learn hierarchical matching patterns over the local interactions. DRMM [3] summarizes the word-level similarities with histogram, which is then combined into a learning-to-rank method. In this paper, two interaction-based models are adopted.

Moreover, there are other ways to improve the performance of IR. Query expansion methods evaluate the given query and expand the query to match additional documents, such as pseudo relevance feedback (PRF) [18–20]. Content-based and collaborative-based filtering methods [21,22] improve the delivery of relevant content to the end-users. Personalization along with relevance feedback techniques can also be employed to increase users' overall satisfaction and quality of experience [23,24].

2.2. Entity-Based Ranking Methods

Recently, several large-scale knowledge graphs such as DBpedia [25] and Freebase [26] have emerged with the rapid development of information extraction methods [27,28]. A knowledge graph is a multi-relational graph composed of entities as nodes and relations as edges [29]. Knowledge graphs contain rich human knowledge about entities and have become very important resources to support other tasks.

To overcome the defects of traditional IR models such as vocabulary mismatch [30], various entity-based models [6,31,32] have been proposed with the aid of entity linking. Bag-of-entities representations use entities to represent texts, and obtain significant improvement over traditional bag-of-words representations [7]. Explicit Semantic Ranking (ESR) [6] uses knowledge graphs and embeddings to take advantage of the semantics. Another way to use knowledge graphs in document-ranking is the word-entity duet framework, which represents queries and documents using both entities and words. As entity-linking tools have not been designed specifically for document-ranking in the previous research, JointSem [33] combines both entity-linking task and entity-based ranking task in an end-to-end model.

Many entity-based models use entity embeddings to correlate queries with documents. Over the past five years, several studies on representation learning could embed a knowledge graph into a continuous vector space while preserving certain information of the graph, such as TransE [34] and TransR [35]. Semantic correlations of entities can be efficiently measured through representation learning. For entities which are semantically related but not matching exactly, embeddings provide a soft match signal to express their similarity. Corresponding to the exact match of query entities, the soft match can also provide signals to information retrieval [6,36].

2.3. Entity Ranking

Traditional entity ranking task always focuses on ranking entities in response to a query in natural language text [37,38], instead of ranking entities directly from documents [39]. However, in this paper, entity ranking is based on the importance of entities in the document without additional data sources. Similarly, NERank [39] also ranks entities without user queries but it is based on document collections, instead of one single document. The task of ranking entities in one document could be applied in other tasks especially for entity-based ranking models. There are two keyword extraction baselines that could be used to rank entities: TextRank and TF-IDF [39]. TF-IDF uses TF and IDF to evaluate how important an entity is to a document in a corpus. TextRank [40] is a graph-based ranking model for text processing, which has already been successfully applied in many natural language processing tasks.

3. Data Analysis

Intuitively, the importance of entities in the document is a key factor in the retrieval task while most models neglect it. When analyzing the results of entity-based models, some mistakes are found due to this factor. To explore the correlation between retrieval results and the importance of entities in the document, data analysis is performed on the ESR dataset (http://boston.lti.cs.cmu.edu/appendices/WWW2016/), which is an academic search dataset used for entity-based ranking. This section describes the ESR dataset, the entity ranking methods used for measuring the importance of entities, and detailed analysis as follows.

3.1. Dataset

The dataset was generated from Semantic Scholar, an academic search engine from Allen Institute. There are 100 queries sampled from Semantic Scholar's query logs and 8541 related papers selected as candidate documents. The academic queries are mainly about computer science (such as 'question answering' and 'deep reinforcement learning'). In addition, there are 5 levels of relevance labels for academic papers in the dataset such as the TREC Web Track: Navigational (4), Exactly Right (3), Relevant (2), Related (1) and Off-Topic (0).

3.2. Entity Ranking

Before exploration, the order of entities in the document should be sorted. Similar to keyword extraction task which extracts important words from documents, entity ranking task is to rank entities according to their importance in the document. Thus, keyword extraction methods can be employed to rank entities in documents. This paper adopts two frequently used baselines, TextRank and TF-IDF, for ranking entities in documents, respectively. Due to their unsupervised character, TextRank and TF-IDF are well-qualified for entity ranking task in this work. The application of two entity ranking methods is aimed to prevent the bias of a single method.

TextRank ranks entities based on the semantic graph extracted from the document. In the graph, document entities are vertices, and co-occurrence relation is used to draw edges between vertices: two entities are connected if their corresponding lexical units co-occur within a window of N words. The score associated with a vertex can be considered as the importance of the corresponding entity:

$$S_i = (1-\alpha) + \alpha * \sum_{j \in C(e_i)} \frac{1}{|C(e_i)|} S_j \tag{1}$$

where S_i is the score of entity e_i, $C(e_i)$ is the set of entities that are linked to e_i, and α is a damping factor.

TF-IDF uses the product of entity frequency tf_i and inverse document frequency idf_i to show the importance of the entity e_i:

$$S_i = tf_i * idf_i \tag{2}$$

The IDF idf_i of entity e_i is counted from Open Research Corpus [41]:

$$idf_i = \log \frac{N}{n_{d_i} + 1} \tag{3}$$

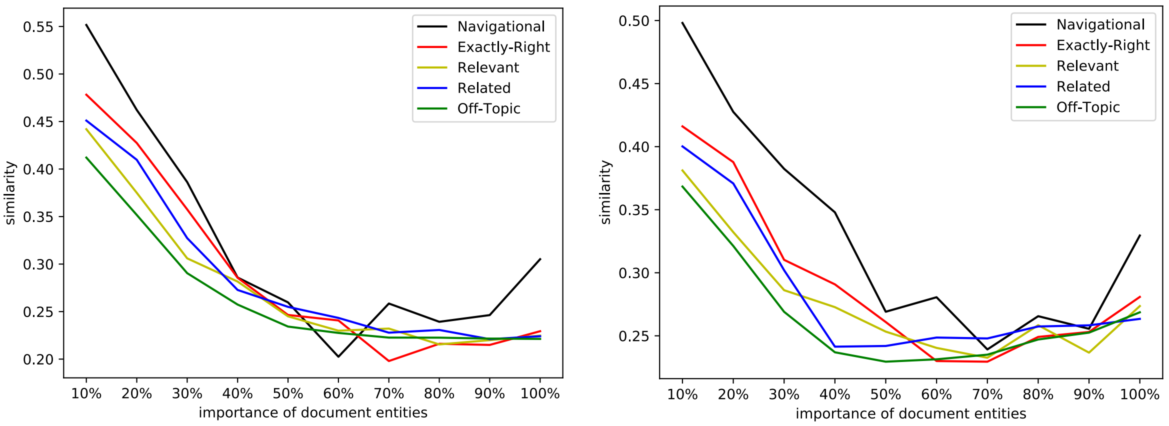
where N represents the total amount of documents in the corpus, and n_{d_i} is the number of documents where the entity e_i appears.

With the help of TextRank or TF-IDF, entities can be sorted in reversed order by their importance and the order of entities can be obtained for the analysis.

3.3. Analysis Result

After ranking entities, this section tries to explore whether the importance of entities is related to document retrieval. The similarities between document entities and query, computed by embeddings (see Section 4 for details), are the essential features for many entity-based models. For document entities, the similarities represent the semantic relatedness to query, which are worth analyzing. Therefore, the correlation between similarities and the order of entities for each relevant level in the dataset is analyzed.

As shown in Figure 2, the X-axis represents the percentage of sorted entities, and the Y-axis represents the average similarity between document entities and corresponding queries. For the average similarity at each percentage for each relevant level, the statistical results of each entity ranking method turn out to be similar: the important entities are always more similar to the query; for the same percentage, the more relevant the documents are, the higher the similarity is. The strong evidence founded in Figure 2 proves that the importance of entities should be considered in retrieval task.



(**a**) (**b**)

Figure 2. The relevance between the order of entities and similarities of document entities: (**a**) Entities are sorted by TextRank; (**b**) Entities are sorted by TF-IDF.

4. Proposed Approach

This section presents the basic models (toy model and ESR) and enhanced ones. According to the analysis in Section 3, the enhanced models integrate the importance of entities into the basic models. Figure 3 presents the framework of the proposed approach. This section first defines the toy model and introduces ESR. In addition, then, toy model and ESR are combined with the order of entities respectively, which is obtained by TextRank or TF-IDF as described in Section 3.2.

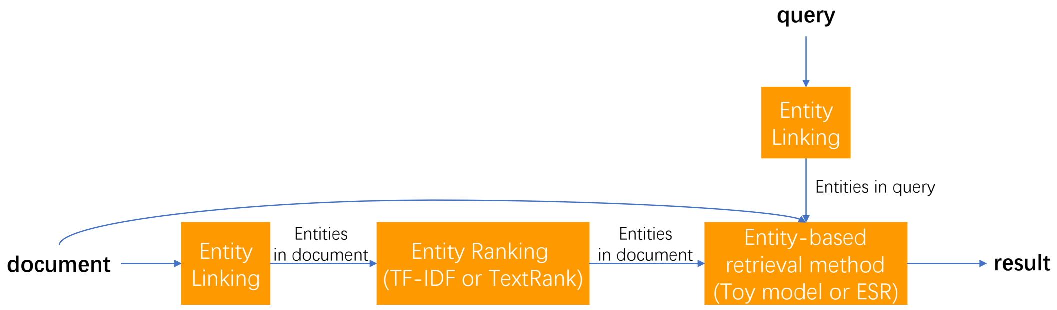


Figure 3. Framework of the proposed approach.

4.1. Basic Entity-Based Models

Toy model and ESR are adopted as the basic models, which rank candidate documents $D = \{d_1, \ldots, d_n\}$ in response to a given query q. The toy model is a simplified method defined by ourselves and ESR is the state-of-the-art method in ESR dataset. Before integration with entity importance, these models are introduced first.

4.1.1. Basic Toy Model

The toy model defined in this paper is a simple interaction-based ranking model which could help us focus on entity importance. Queries and documents are represented by the entities contained in them. The toy model uses the average similarity between query entities and document entities as the ranking score of the document.

Entity-Based Representation

In toy model, the representations of query q and document d are generated by entity linking. After that, toy model gets the representation of the query q (or the document d): vector $\overrightarrow{E_q}$ (or $\overrightarrow{E_d}$), where each dimension refers to an entity e in q (or d) and the weight is the frequency of this entity appearing in the query or document.

The Generation of Query-Document Entity Translation Matrix

For characterizing the similarity between the query and document, the query-document entity translation matrix $\mathbf{M}^{n \times m}$ is generated through the representations of q and d, where n is the size of $\overrightarrow{E_d}$ and m is the size of $\overrightarrow{E_q}$. Each element in the matrix is the cosine similarity between embeddings of entity e_q in query q and entity e_d in document d:

$$s = cos(V(e_q), V(e_d)) \tag{4}$$

where $V(e_q)$ and $V(e_d)$ denote the embeddings of entities e_q, e_d respectively. In the matrix, each row corresponds to an entity in document d and each column corresponds to an entity in query q.

Different from the exact matching in traditional IR, the query-document entity translation matrix provides a soft matching between query entity and document entity where the cosine similarity remains less than 1.

Document Ranking

The relevance score of document d corresponding to query q can be computed by the average similarity in the query-document entity translation matrix $\mathbf{M}^{n \times m}$. As a formula, the score of document d can be presented as

$$f(q, d) = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} \tag{5}$$

where s_{ij} is the element in $\mathbf{M}^{n\times m}$ and $f(q,d)$ is the relevance score between document d and query q. Although the toy model is simple, it is very practical in some cases.

4.1.2. Basic Explicit Semantic Ranking (ESR)

As other entity-based retrieval models [7,31,42], ESR extracts interactive information between document entities and query entities to generate ranking features. Firstly, ESR generates entity-based representation and query-document entity translation matrix $\mathbf{M}^{n\times m}$ as described in Section 4.1.1. Then, ESR uses max-pooling and bin-pooling step to generate ranking features for learning-to-rank model.

Max-Pooling

After query-document entity translation matrix generation as in Section 4.1.1, the score of each entity in document d is calculated by the matrix.

Query level max-pooling is executed on the translation matrix $\mathbf{M}^{n\times m}$, aiming to find out the most similar entity to each document entity. For each row in $\mathbf{M}^{n\times m}$, max-pooling figures out the maximum element r_i in this row:

$$r_i = max(s_{i1}, \dots, s_{im}) \tag{6}$$

where $s_{i1}, \dots, s_{im}$ are the elements in one row. The maximum element of the document entity e_d can be considered as the score of this entity. In addition, the vector $\mathbf{R}^n = (r_1, \dots, r_n)$ created by the max-pooling reflects the similarity between document entities and query q.

Learning to Rank with Bins

To transform the variable-length vector $\mathbf{R}^n$ into a fixed-length matching histogram, ESR adopts a bin-pooling step which generates different features according to the similarity. The features are used in learning-to-rank model to compute the final score of the document d to query q.

Bin-pooling represents document d as fixed-length which groups entities according to their values:

$$B_k(q,d) = log \sum_{\mathbf{r}_i \in \mathbf{R}^n} I(st_k \le r_i \le ed_k) \tag{7}$$

where $\mathbf{R}^n$ is the vector generated by max-pooling, $[st_k, ed_k)$ is the range for the kth bin, and $B_k(q,d)$ represents the number of document entities whose scores are in the corresponding bin. ESR discretizes the range of the scores of document entities into a set of ordered bins, and treats exact matching as a separate bin, so that both exact matching and soft-matching can be used by ESR.

Learning-to-rank model [43] is performed by using bin scores as features to generate the ranking function:

$$f(q,d) = \mathbf{W}^T B(q,d) \tag{8}$$

where $B(q,d)$ is the bin scores, W is the parameters needed to learn in ESR, and $f(q,d)$ represents the final score of query-document pair (q,d).

4.2. Enhanced Entity-based Model by Considering the Importance of Entities

Obviously, these basic entity-based models mainly regard similarity between entities as major foundation whereas do not employ the importance of entities in the document. This issue is solved by allotting proper weights to entities. On the basis of Section 4.1, entities in each document could be divided into three categories in terms of the order calculated by TextRank or TF-IDF: the first third of the ranked entities are important for the document; the middle third of the ranked entities are related with the document; and the last third of the ranked entities are more likely to be irrelevant to the document. These categories are distributed by diverse weights ω_1, ω_2 and ω_3 respectively with the constraint $\omega_1 \ge \omega_2 \ge \omega_3 \ge 0$. The following describes relevant details about how to take advantage of entity importance.

4.2.1. Enhanced Toy Model by Considering the Importance of Entities

As Figure 4 illustrates, the enhanced toy model combines both similarity and the importance of entities to recalculate the scores of entities.

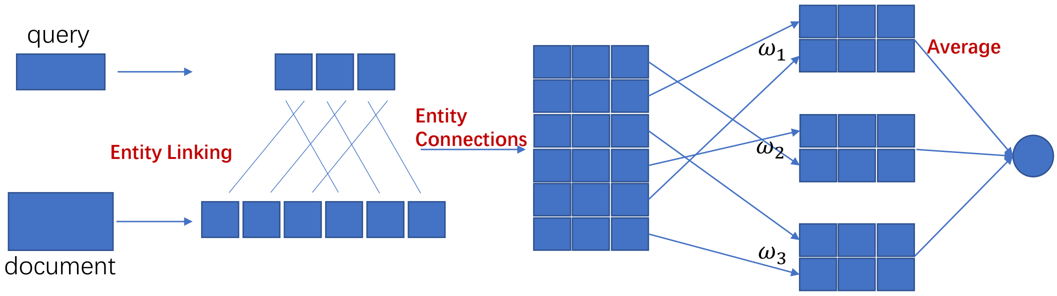


Figure 4. An overview of enhanced toy model.

Important entities should contribute a greater amount to the ranking, and in contrast noise from irrelevant entities should be reduced. By multiplied with different weights, scores of different entities will be more reasonable:

$$v_{ij} = \omega_l * s_{ij} \tag{9}$$

where s_{ij} is the element in query-document entity translation matrix $\mathbf{M}^{n\times m}$, ω_l is the weight dependent on the order of entity e_i in document, and v_{ij} is the optimized score presenting the correlation between document entity i and query entity j. The final ranking score is then substituted in a more sensible way:

$$f(q,d) = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij} \tag{10}$$

The improvement in the succinct model could reflect the effectiveness of entity importance definitely.

4.2.2. ESR optimized by considering the importance of entities

Figure 5 illustrates how the ESR is optimized.

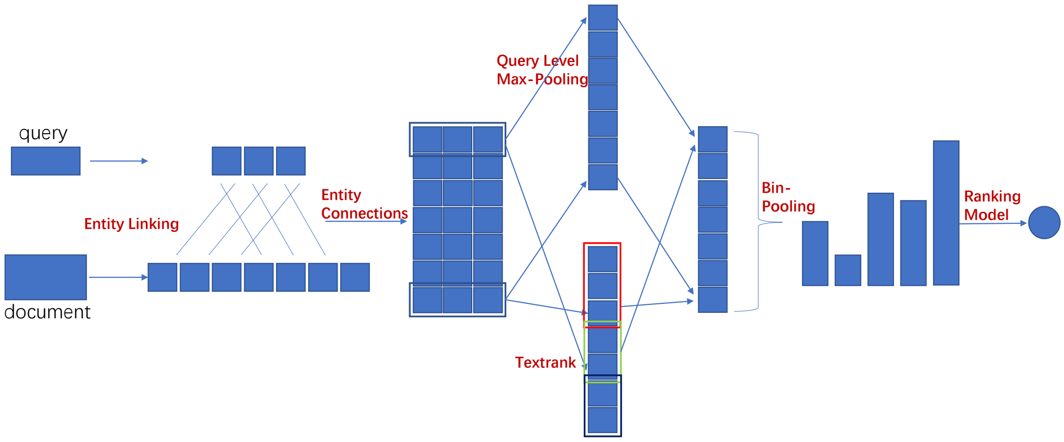


Figure 5. An overview of enhanced ESR.

As described in Section 4.1.2, the vector $\mathbf{R}^n = (r_1, \ldots, r_n)$ created by the max-pooling shows the similarity between document and query q. Subsequently, the score of document entity is optimized by the following function:

$$v_i = \omega_l * r_i \tag{11}$$

where r_i is the similarity corresponding to the l-th row in $\mathbf{M}^{n\times m}$, and ω_l depends on the order of this entity, and v_i is the score of the entity in the document towards query q. The vector $\mathbf{V}^n = (v_1, \ldots, v_n)$ created by $\mathbf{R}^n$ and TextRank (or TF-IDF) reflects the relevance between document entities and query q. Then the bin-pooling operated on $\mathbf{V}^n$ summarizes more sufficient ranking features than basic ESR by considering the importance of entities. Learning-to-rank model performed on ranking features figures out the final ranking result of each document.

5. Results and Discussion

The experiments investigate the effectiveness of entity importance. This section provides evaluation results and the discussion of the impact of the importance of entities. Both TextRank and TF-IDF are tested to rank entities for entity-based models. Then the performance on different scenarios is discussed further. To verify the effectiveness of entities ranking, only the top 1/3 entities in the ranking are used to optimize the performance of ESR.

5.1. Experimental Setup

To evaluate the proposed models, a set of experiments are conducted using the datasets of ESR [6] described in Section 3. The entities contained in queries and documents are identified by Tagme software [44], which is an excellent entity-linking system. The experiments concentrate on the entities from the document's title and abstract, which contain adequate information of a document.

The baselines of these experiments are tf.idf-F, toy model and the ESR. Tf.idf-F is a widely used retrieval method; toy model and ESR are entity-based retrieval models introduced in Section 4.1. In ESR model, the score from Semantic Scholar's production system is not used because ESR and the enhanced models can be applied in not only Semantic Scholar but also other retrieval systems. The embeddings of entities in ESR and toy model are generated from Wikidata (https://www.wikidata.org) representation via TransE [34] with the help of OpenKE (http://openke.thunlp.org/index/about). A popular approach, RankSVM [45], is served as the learning-to-rank model in the experiments.

In the experiments, the co-occurrence window-size of TextRank is set to five words. Same as ESR, the negative scores of entities are discarded as they are not informative, so the effective scores of entities are distributed in the range $[0,1]$. Moreover, the interval $[0,1]$ is discretized into five bins including an exact matching bin: $[1,1]$, $[0.75,1)$, $[0.50,0.75)$, $[0.25,0.50)$ and $[0,0.25)$. Ten features are generated for RankSVM by bin-pooling on title and abstract. Both baselines and the models integrated the importance of entities are trained and tested using ten-fold cross validation and the 'c' of RankSVM is selected from the set $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$.

5.2. Evaluation and Analysis

In the following, the enhanced toy model and ESR model with TextRank and TF-IDF are evaluated. All methods are evaluated by normalized discounted cumulative gain at rank 20 (NDCG@20), which is the main evaluation metric in the TREC Web Track, and evaluated by NDCG@1, NDCG@5, NDCG@10, P@3 and P@10.

5.2.1. Results of Enhanced Toy Model

First, the enhanced toy model is compared with baselines. Table 1 presents the experimental results obtained from tf.idf-F, toy model and the enhanced models. Although toy model is very simple, the performance of it turns out to better than tf.idf-F on NDCG@10 and NDCG@20, which proves the effectiveness of the entity-based model.

Table 1 demonstrates that, both the toy model optimized by TextRank and the toy model optimized by TF-IDF get improvement on all evaluation metrics. The enhanced models even outperform basic model by over 10% on NDCG@1 and NDCG@5. **W/T/L** are the number of queries a model improves (win), does not change (tie), and hurts (loss), compared with toy model on NDCG@20. The **W/T/L** values in Table 1 demonstrate that both enhanced toy models can win more cases.

Table 1. Retrieval performance of tf.idf-F, toy model and enhanced toy model on NDCG@K. The performance of proposed models is compared with toy model in percentages. The number of queries a model improves (**W**in), does not change (**T**ie) or hurt (**L**oss), compared with ESR, are provided in the **W/T/L** column. The best results in each metric are marked bold.

Method	NDCG@1		NDCG@5		NDCG@10		NDCG@20		W/T/L
tf.idf-F	0.2020	+12.98%	0.2254	+3.73%	0.2741	-2.94%	0.3299	-16.73%	30/2/68
toy model	0.1788	-	0.2173	-	0.2824	-	0.3962	-	-
toy model with TextRank	**0.2221**	+24.22%	0.2402	+10.54%	**0.3025**	+7.12%	0.4108	+3.69%	49/18/33
toy model with TF-IDF	0.2128	+19.02%	**0.2410**	+10.91%	0.2981	+5.56%	**0.4143**	+4.57%	45/18/37

To further evaluate the enhanced models, the P@3 and P@10 are provided to measure how many related documents are among the top 3 and 10 positions. In current IR models, queries have many related documents, but few users will read all of them. Users usually pay more attention to the top documents returned from IR systems (for example, top 10 documents which are the related results on the first search results page), therefore the earlier positions are more important for user satisfaction. It can be seen from the data in Table 2 that both enhanced models also outperform toy model on P@3 and P@10, which demonstrates the enhanced toy models could improve user satisfaction.

Table 2. Retrieval performance of toy model and enhanced toy models on P@3 and P@10. The best results in each metric are marked bold.

Method	P@3		P@10	
toy model	0.4167	-	0.4400	-
toy model with TextRank	**0.4667**	+12.00%	**0.4510**	+2.50%
toy model with TF-IDF	0.4500	+7.99%	0.4500	+2.72%

5.2.2. Results of Enhanced ESR

As shown in Table 3, the results of ESR are significantly higher than vector space model especially for NDCG@20, although the scores from Semantic Scholar's production system are not employed. These results further verify the effectiveness of entity-based model and soft-matching as analyzed in the ESR [6].

Table 3. Retrieval performance of tf.idf-F, basic ESR and enhanced ESR on NDCG@K. The performance of enhanced models is compared with ESR in percentages. The number of queries a model improves (**W**in), does not change (**T**ie) or hurt (**L**oss), compared with ESR, are provided in the **W/T/L** column. The best results in each metric are marked bold.

Method	NDCG@1		NDCG@5		NDCG@10		NDCG@20		W/T/L
tf.idf-F	0.2020	−2.88%	0.2254	−1.79%	0.2741	-6.26%	0.3299	−17.21%	28/5/67
ESR	0.2080	-	0.2295	-	0.2924	-	0.3985	-	-/-/-
ESR with TextRank	**0.2270**	+9.13%	**0.2358**	+2.75%	**0.3029**	+3.59%	**0.4094**	+2.74%	46/23/31
ESR with TF-IDF	0.2154	+3.56%	0.2319	+1.05%	0.2949	+0.85%	0.4040	+1.38%	40/25/35

Although ESR is the state-of-the-art model on this dataset, the ESR model optimized by TextRank still performs better than ESR for all evaluation metrics. On NDCG@1, the improvement is even higher than 9%. Compared with the result of ESR, the enhanced model using TextRank improves 46 queries' NDCG@20, unchanges 23 queries and only hurts 31 queries. The ESR model optimized by TF-IDF also have improvement on all evaluation metrics. The performances of the ESR model optimized by TextRank (or TF-IDF) are also stronger on P@3 and P @10 as Table 4 shown.

Table 4. Retrieval performance of ESR and enhanced ESR models on P@3 and P@10. The best results in each metric are marked bold.

Method	**P@3**		**P@10**	
ESR	0.4600	-	0.4550	-
ESR with TextRank	0.4633	+0.72%	0.4650	+2.20%
ESR with TF-IDF	**0.4900**	+6.52%	**0.4690**	+3.08%

These experiments provide strong evidence that considering the importance of entities could improve the performance of entity-based retrieval models. The results on both toy model and ESR model exclude accidental factors. The result of TF-IDF is not as well as TextRank in the experiments. Maybe this is the reason the result of the enhanced model using TF-IDF is not very well.

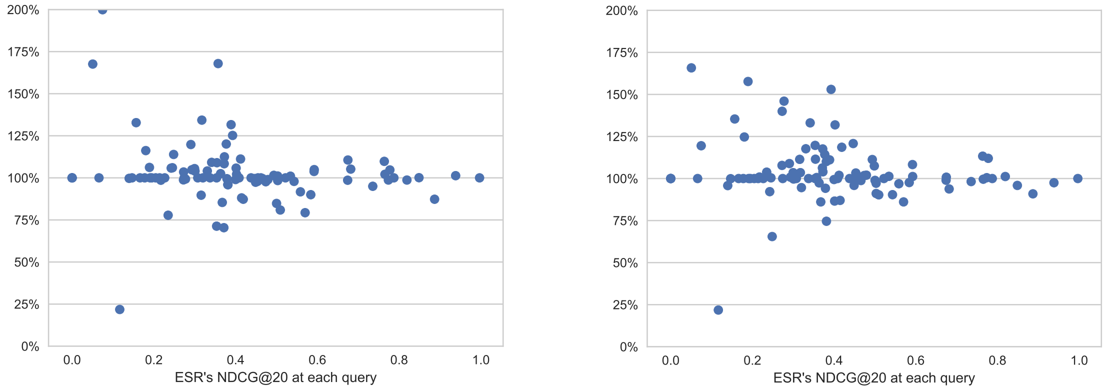
Besides, compared with the basic models, the additional implementation cost of the enhanced models is mainly the computation of the importance of entities in documents. The execution time measurements of TextRank and TF-IDF are performed on 2.8 GHz Intel Core i7 GPU and 8 GB of RAM. Execution times were measured on 1000 randomly chosen documents and each experiment was repeated 10 times. TF-IDF only costs 0.202 s to process 1000 documents, and the average execution time of TextRank on one document is 0.121 s. Moreover, the importance of entities can be measured off-line, so that the additional implementation cost does not affect the efficiency of retrieval. Therefore, the enhanced models could get improvement with less implementation cost.

5.3. Performance on Different Scenarios

This section analyzes the influence of the importance of entities in two different scenarios: multiple difficulty degrees and multiple query length degrees. The experiments compare the strong baseline, ESR, with its enhanced models on these scenarios, respectively.

5.3.1. Multiple Difficulty Degrees

This part explores the performance of the enhanced models on queries at different difficulty. Figure 6 compares the performance of the enhanced models on individual queries with ESR. Each point in Figure 6 corresponds to a query. The value on the X-axis is the ESR's NDCG@20 of the query, and the value on the Y-axis is the enhanced model's relative NDCG@20 (percentage) compared with ESR. Those points above the 100% line represent the queries improved by the enhanced model.



(**a**) the ESR model optimized by TextRank

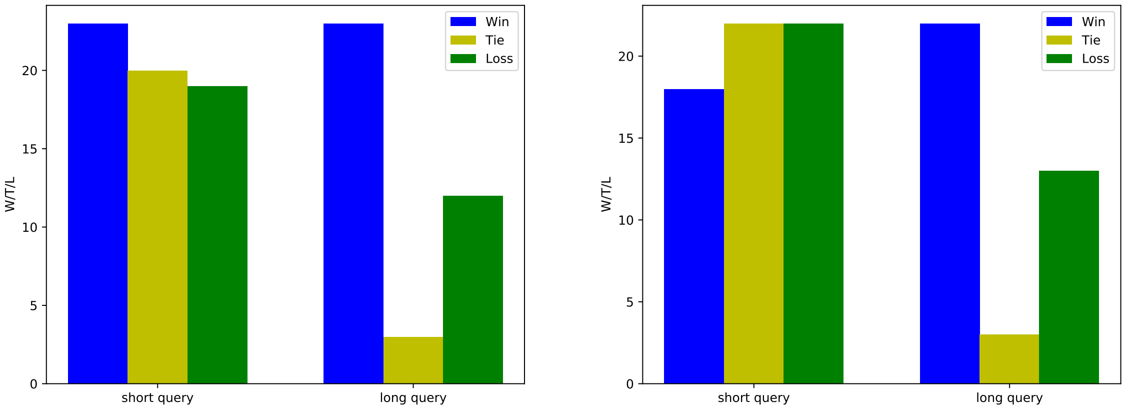
(**b**) the ESR model optimized by TF-IDF

Figure 6. The enhanced model's relative NDCG@20 compared with ESR on individual queries. Each point corresponds to a query in the ESR dataset. The X-axis marks the ESR's NDCG@20 of the query, and the value on the Y-axis is the enhanced model's relative NDCG@20 (percentage) compared with ESR.

It can be seen from these scatters that the enhanced models primarily improve the queries that are hard for ESR (NDCG@20 <= 0.4): for the ESR model optimized by TextRank, the performance improvement on these queries is more than 6.9%; for the ESR model optimized by TF-IDF, the gain of NDCG@20 is also about 6.3%. On the queries that are easy for ESR (NDCG@20 > 0.4), the improvement of the enhanced models is relatively small. Moreover, for the queries that the enhanced model improves, many of them obtain huge improvement by considering the importance of entities. In addition, for the queries that the enhanced model hurt, there is only a small performance penalty for most of them.

5.3.2. Multiple Query Length Degrees

This section further evaluates the effectiveness of the importance of entities on short queries (one entity) and long queries (more than one entity). The results on short queries and long queries are shown in Figure 7 respectively. Both enhanced models have more win cases on long queries and could pay more attention to the important entities in long queries.



(**a**) the ESR model optimized by TextRank (**b**) the ESR model optimized by TF-IDF

Figure 7. The **W/T/L** on short queries (only one entity) and long queries (more than one entity). The X-axes mark two query levels, and Y-axes are the **W/T/L** values.

These two experiments show that the enhanced models mainly improve the performance of longer queries and the queries where ESR fails.

5.4. Impact of Document Entities with Different Order

To further confirm the effectiveness of the importance of entities, this section explores the impact of document entities with different orders. As shown in Figure 8, as in the method in Section 4.2, the document entities are divided into three groups according to their importance: important entities, related entities, and irrelevance entities. ESR using different groups entities is tested to analyze the contribution of each group.

Firstly, this work only uses the important entities in ESR model. In other words, the bin features of important entities are multiplied by a coefficient $\omega_l = 1$, and the bin features of other entities are multiplied by coefficient $\omega_2 = 0$ and $\omega_3 = 0$. As shown in Table 5, the model with only the important entities also outperforms ESR in all evaluation metrics. This indicates that the first third of ranked entities contain enough information for IR and important entities play a crucial role in ranking task. The ranking excludes some interference like wrongly linked entities.

Then the important entities and related entities are adopted in ESR model, which means that ω_3 is 0 while ω_l and ω_2 are hyper-parameters to be selected. Table 5 shows that the impact of related entities is limited although the coefficient ω_2 is optimized.

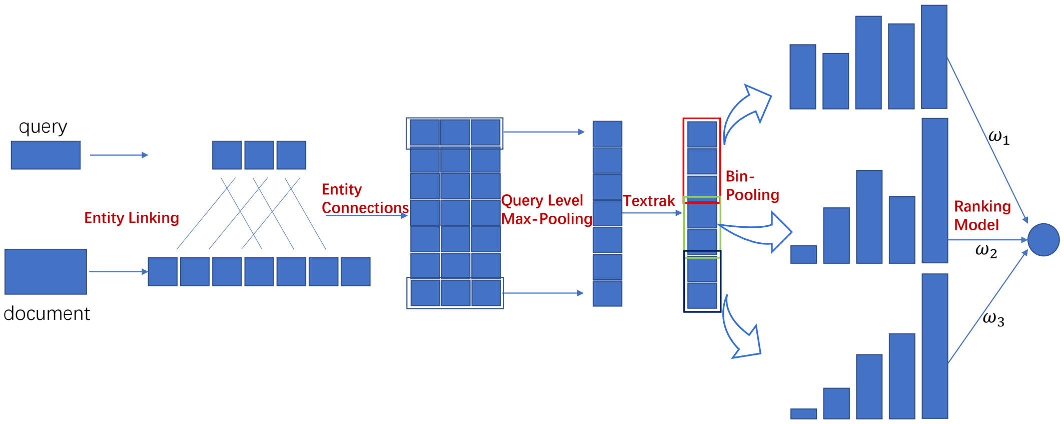


Figure 8. The architecture of the analysis model.

This work also uses all the document entities in ESR model and ω_l, ω_2, ω_3 are hyper-parameters to be selected. The third group of entities is sometimes from the wrong linked entities. The results show that the irrelevant entities are ineffective for ESR or can even be harmful.

Table 5. Performance of ESR with different parts of entities. The best results in each metric are marked bold.

Method	NDCG@1	NDCG@5	NDCG@10	NDCG@20
important entities	**0.2252**	0.2427	**0.3042**	0.4070
important entities + related entities	0.2203	**2438**	0.3014	0.4072
all entities	0.2244	0.2397	0.2984	**0.4089**

Through this experiment, it can be found that the more important the entities are, the more impact the entities have. The experiment also confirms that the importance of entities should be considered in ESR model.

6. Conclusions

This paper points out that the importance of entities in the document is a crucial factor in IR, while many existing entity-based models have not taken it into account. Thus, this paper explores the impact of the importance of entities from two aspects. First, data analysis is carried out on the ESR dataset and reveals the correlation between document-ranking and the importance of entities. Then this paper attempts to optimize the entity-based models with the importance of entities considered. TextRank and TF-IDF are designed to measure the importance of document entities, and the enhanced toy models and ESR models are proposed to make use of the importance of entities. Two IR models and two entity ranking models provide evidence for the effectiveness.

Extensive experiments on the ESR dataset reveal the effectiveness of the enhanced models. Statistically significant improvements over the basic model are observed on all evaluation metrics, especially for the toy model optimized by TextRank, which gets 24.22% increase on NDCG@1, and 12% increase on P@3. Besides, assigning different weights to entities according to their importance greatly improves the effectiveness on long queries and the queries where ESR fails. Furthermore, the research has shown that the important entities in the document are also essential for IR. While the enhanced models get improvement on all evaluation metrics, the additional implementation cost is quite low. We hope that our study on the importance of entities will inspire research into more in-depth entity-based retrieval models in the future.

Author Contributions: Z.L. and X.L. conceived and designed the algorithms and performed the experiments and analyzed the results; Z.L., X.L. and D.Z. wrote and revised the manuscript; G.X. and F.L. discussed the data and corrected the manuscript; G.X. and L.W. provided relevant information and instructions during the design of experiments. All authors have read and approved the final manuscript.

Funding: This work was supported by the Foundation of Precision Poverty Alleviation Technology under Grant 42-Y30B12-9001-17/18.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, J.; Fan, Y.; Ai, Q.; Croft, W.B. Semantic Matching by Non-Linear Word Transportation for Information Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*; ACM: New York, NY, USA, 2016; pp. 701–710.
2. Huang, P.S.; He, X.; Gao, J.; Deng, L.; Acero, A.; Heck, L. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*; ACM: New York, NY, USA, 2013; pp. 2333–2338.
3. Guo, J.; Fan, Y.; Ai, Q.; Croft, W.B. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*; ACM: New York, NY, USA, 2016; pp. 55–64.
4. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: New York, NY, USA, 2008.
5. Robertson, S.E.; Zaragoza, H. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* **2009**, *3*, 333–389. [CrossRef]
6. Xiong, C.; Power, R.; Callan, J. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In Proceedings of the 26th International Conference on World Wide Web; International World Wide Web Conferences Steering Committee: Geneva, Switzerland, 2017; pp. 1271–1279.
7. Xiong, C.; Callan, J.; Liu, T.Y. Bag-of-Entities Representation for Ranking. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*; ACM: New York, NY, USA, 2016; pp. 181–184.
8. Liu, Z.; Xiong, C.; Sun, M.; Liu, Z. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 2395–2405.
9. Ferragina, P.; Scaiella, U. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*; ACM: New York, NY, USA, 2010; pp. 1625–1628.
10. Hasibi, F.; Balog, K.; Bratsberg, S.E. Entity Linking in Queries: Tasks and Evaluation. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*; ACM: New York, NY, USA, 2015; pp. 171–180.
11. Yin, D.; Hu, Y.; Tang, J.; Daly, T.; Zhou, M.; Ouyang, H.; Chen, J.; Kang, C.; Deng, H.; Nobata, C.; et al. Ranking Relevance in Yahoo Search. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp. 323–332.
12. Liu, X.; Fang, H. Latent entity space: A novel retrieval approach for entity-bearing queries. *Inf. Retr. J.* **2015**, *18*, 473–503. [CrossRef]
13. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Lake Tahoe, Nevada, 2013; pp. 3111–3119.
14. Li, H.; Xu, J. Semantic Matching in Search. *Found. Trends Inf. Retr.* **2014**, *7*, 343–469. [CrossRef]
15. Shen, Y.; He, X.; Gao, J.; Deng, L.; Mesnil, G. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. In *Proceedings of the 23rd International Conference on World Wide Web*; ACM: New York, NY, USA, 2014; pp. 373–374.
16. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Lake Tahoe, Nevada, 2014; pp. 2042–2050.

17. Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; Cheng, X. Text Matching as Image Recognition. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2793–2799.
18. Buckley, C.; Salton, G.; Allan, J.; Singhal, A. Automatic Query Expansion Using SMART: TREC 3. In Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, MD, USA, 2–4 November 1994; pp. 69–80.
19. Cao, G.; Nie, J.Y.; Gao, J.; Robertson, S. 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*; ACM: New York, NY, USA, 2008; pp. 243–250.
20. Metzler, D.; Croft, W.B. Latent Concept Expansion Using Markov Random Fields. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2007; pp. 311–318.
21. Balabanović, M.; Shoham, Y. Fab: Content-based, Collaborative Recommendation. *Commun. ACM* **1997**, *40*, 66–72. [CrossRef]
22. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. *Knowl.-Based Syst.* **2013**, *46*, 109–132. [CrossRef]
23. Pouli, V.; Kafetzoglou, S.; Tsiropoulou, E.E.; Dimitriou, A.; Papavassiliou, S. Personalized multimedia content retrieval through relevance feedback techniques for enhanced user experience. In Proceedings of the 2015 13th International Conference on Telecommunications (ConTEL), Graz, Austria, 13–15 July 2015; pp. 1–8.
24. Stai, E.; Kafetzoglou, S.; Tsiropoulou, E.E.; Papavassiliou, S. A holistic approach for personalization, relevance feedback & recommendation in enriched multimedia content. *Multimedia Tools Appl.* **2018**, *77*, 283–326.
25. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*; Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
26. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*; ACM: New York, NY, USA, 2008; pp. 1247–1250.
27. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 260–270.
28. Shen, W.; Wang, J.; Han, J. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 443–460. [CrossRef]
29. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
30. Croft, W.B.; Metzler, D.; Strohman, T. *Search Engines—Information Retrieval in Practice*; Pearson Education: London, England, 2009.
31. Raviv, H.; Kurland, O.; Carmel, D. Document Retrieval Using Entity-Based Language Models. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2016; pp. 65–74.
32. Dietz, L.; Kotov, A.; Meij, E. Utilizing Knowledge Bases in Text-centric Information Retrieval. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*; ACM: New York, NY, USA, 2016.
33. Xiong, C.; Liu, Z.; Callan, J.; Hovy, E. JointSem: Combining Query Entity Linking and Entity Based Document Ranking. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*; ACM: New York, NY, USA, 2017; pp. 2391–2394.
34. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Lake Tahoe, Nevada, 2013; pp. 2787–2795.
35. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.

36. Xiong, C.; Dai, Z.; Callan, J.; Liu, Z.; Power, R. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2017; pp. 55–64.
37. De Vries, A.P.; Vercoustre, A.M.; Thom, J.A.; Craswell, N.; Lalmas, M. *Overview of the INEX 2007 Entity Ranking Track*; Focused Access to XML Documents; Fuhr, N., Kamps, J., Lalmas, M., Trotman, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 245–251.
38. Kang, C.; Yin, D.; Zhang, R.; Torzec, N.; He, J.; Chang, Y. Learning to rank related entities in Web search. *Neurocomputing* **2015**, *166*, 309–318. [CrossRef]
39. Wang, C.; Zhou, G.; He, X.; Zhou, A. NERank+: A graph-based approach for entity ranking in document collections. *Front. Comput. Sci.* **2018**, *12*, 504–517. [CrossRef]
40. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Texts. In *Emnlp*; Association for Computational Linguistics: Stroudsburg, Pennsylvania, 2004; pp. 404–411.
41. Ammar, W.; Groeneveld, D.; Bhagavatula, C.; Beltagy, I.; Crawford, M.; Downey, D.; Dunkelberger, J.; Elgohary, A.; Feldman, S.; Ha, V.; et al. Construction of the Literature Graph in Semantic Scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*; Association for Computational Linguistics: Stroudsburg, Pennsylvania, 2018; pp. 84–91.
42. Xiong, C.; Callan, J.; Liu, T.Y. Word-Entity Duet Representations for Document Ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2017; pp. 763–772.
43. Liu, T.Y. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* **2009**, *3*, 225–331. [CrossRef]
44. Ferragina, P.; Scaiella, U. Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE Softw.* **2012**, *29*, 70–75. [CrossRef]
45. Joachims, T. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2002; pp. 133–142.