



HAL
open science

DSRIM: A Deep Neural Information Retrieval Model Enhanced by a Knowledge Resource Driven Representation of Documents

Gia-Hung Nguyen, Laure Soulier, Lynda Tamine, Nathalie Bricon-Souf

► **To cite this version:**

Gia-Hung Nguyen, Laure Soulier, Lynda Tamine, Nathalie Bricon-Souf. DSRIM: A Deep Neural Information Retrieval Model Enhanced by a Knowledge Resource Driven Representation of Documents. 3rd International Conference on the Theory of Information Retrieval (ICTIR 2017), ACM SIGIR, Oct 2017, Amsterdam, Netherlands. pp.19-26, 10.1145/3121050.3121063 . hal-01549543

HAL Id: hal-01549543

<https://hal.sorbonne-universite.fr/hal-01549543>

Submitted on 28 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DSRIM: A Deep Neural Information Retrieval Model Enhanced by a Knowledge Resource Driven Representation of Documents

Gia-Hung Nguyen

IRIT, Université de Toulouse, CNRS
Toulouse, France
gia-hung.nguyen@irit.fr

Lynda Tamine

IRIT, Université de Toulouse, CNRS
Toulouse, France
tamine@irit.fr

Laure Soulier

Sorbonne Universités, UPMC Univ Paris 06 CNRS, LIP6
UMR 7606, 4 place Jussieu 75005 Paris, France
laure.soulier@lip6.fr

Nathalie Bricon-Souf

IRIT, Université de Toulouse, CNRS
Toulouse, France
nathalie.souf@irit.fr

ABSTRACT

Tackling the vocabulary mismatch has been a long-standing and major goal in information retrieval. The state-of-the-art solutions mainly rely on leveraging either the relational semantics provided by external resources or the distributional semantics, recently investigated by deep neural approaches. Guided by the intuition that the relational semantics might improve the effectiveness of deep neural approaches, we propose the Deep Semantic Resource Inference Model (DSRIM) that relies on a two-fold contribution: 1) a representation of raw-data that models the relational semantics within text representations by jointly considering objects and relations expressed in a knowledge resource, and 2) an end-to-end neural architecture that jointly learns the query-document relevance and the combined distributional and relational semantic representation of documents and queries. The experimental evaluation is carried out on two TREC datasets from TREC Terabyte and TREC CDS tracks, and two different knowledge resources, respectively WordNet and MeSH. The results indicate that knowledge resource-driven representations allow obtaining similar representations for similar documents while discriminating non-similar documents. Also, we show that our model outperforms state-of-the-art semantic and deep neural information retrieval models.

CCS CONCEPTS

•Information systems → Retrieval models and ranking;
•Computing methodologies → Semantic networks; Neural networks;

KEYWORDS

Ad-hoc IR, knowledge resource, semantic document representation, deep neural architecture

1 INTRODUCTION

Tackling the vocabulary mismatch has been a long-standing and major goal in information retrieval (IR). To infer and match discrete word senses within the context of documents and queries

being matched, one line of work makes use of hand-labelled external knowledge resources. The latter can be classified into two categories: 1) those representing linguistic knowledge, either general (e.g., WordNet) or domain-oriented (e.g., UMLS), and 2) those, known as knowledge graphs (e.g., DBpedia, Freebase), representing factual information about entities and semantic relations between entities. In IR, such resources allow to exploit objects and their relations (e.g., synonymy, hyperonymy) within, e.g., query or document expansion [1, 36] to lower the vocabulary mismatch between queries and documents; this is referred to as the *relational semantics*.

Another line of work attempts to automatically infer hidden word senses from corpora using word collocations by performing dimensionality reduction techniques [8]. Based on this approach, there is a recent growth interest toward learning algorithms that map words and their contexts into a lower dimensional dense vector space [21, 29]. The use of such latent representations through deep neural networks for supporting a search task has been the focus of recent work [10, 12, 30]. Although those deep IR models have been evaluated on large datasets (e.g., search logs of commercial search engines), the learning of a relevance function, and accordingly of latent representations from plain text, suffers from several limitations: 1) tackling traditional IR models, such as BM25 or language models, remains a difficult task [10–12], 2) learning the relevance function on full text does not allow the network convergence, leading to focus on a query-document title matching [10, 12, 30], and 3) latent learned representations of word senses from corpora are unable to model distinct meanings and to convey meanings provided by existing semantic resources [13]. Recently, several authors investigated learning the relevance model from plain text by modeling documents and queries through local interactions of terms [10, 11]. The authors [10] also argue that there is a crucial need to build deep learning models satisfying both the matching task based on the semantics of text as usually done in natural language processing (NLP), and the exact matching task rather adapted to ad-hoc IR.

Guided by the intuition that the relational semantics could improve the effectiveness of deep neural IR models [24], we investigate how to leverage both knowledge resources and deep learning approaches to perform ad-hoc IR.

This is the author's pre-print version of the work. It is posted here for your personal use, not for redistribution. The definitive version will be published in Proceedings of ICTIR 2017

To the best of our knowledge, this is one of the first approach combining the distributional and the relational semantics of documents and queries with the goal to jointly enhance 1) the learning of text representation obtained in a low-dimensional common semantic feature space and 2) the underlying ranking function. More particularly, the paper contributions are twofold:

- A *Deep Semantic Resource Inference Model (DSRIM)* combining the distributional and the relational semantics of documents and queries through:

- A *knowledge resource-driven representation*, modeling the relational semantics of text at the raw data level using knowledge resources. More particularly, the premise of our representation relies on two assumptions: A1) a text is a bag of identified objects from a knowledge resource, and A2) semantically similar texts are deemed to entail similar/related objects. To do so, we present a method consistent with both assumptions that jointly projects objects and their relations in a unique vector. To deal with a large number of object-to-object relations (e.g., word-to-word or concept-to-concept) in the knowledge resource, we propose the *relation mapping* method that aims at projecting pairs in a low-dimensional space of object clusters. Our method is flexible since it can be used with any resource providing objects and relations between objects.

- An end-to-end neural network which learns an enhanced document-ranking function using input vectors combining both the distributional and the relational semantic representations of document/query. While the first representation is modeled using the *ParagraphVector* model [2], well-known for capturing complex information structure in plain text, the second representation refers to our proposed knowledge resource-driven representation.

- A *thorough experimental evaluation* aiming at assessing the quality of the knowledge resource-driven representation and the effectiveness of our DSRIM model. Our evaluation is carried out on two TREC datasets, namely TREC PubMed CDS and TREC GOV2 Terabyte, and two widely used knowledge resources, respectively MeSH¹ and WordNet². It is worth mentioning that, unlike previous contributions [12, 30] experimentally evaluated on document titles, our experiments are performed using full-texts.

The rest of this paper is organized as follows. After reviewing the related work in Section 2, we motivate our contribution and present the research questions in Section 3. We describe in Section 4 the DSRIM model. Section 5 details the datasets and the evaluation methodology. Section 6 presents and discusses the experimental results. Section 7 concludes the paper and introduces perspectives.

2 RELATED WORK

2.1 On the Semantic Representation of Words, Documents, Objects, and Relations.

The potential of word semantic representations learned through neural approaches has been introduced in [21, 29], opening several perspectives in NLP and IR tasks. Beyond word embeddings, several work focuses on the representation of sentences [22], documents [16, 32], and also objects and relations expressed in knowledge resources [5, 14, 20, 33]. Within the latter, most of work investigates

the representation of objects and relations on the basis of object-relation-object triplets. The main principle has been introduced in the TransE model [5], relying on the assumption that the embedding of object o_j should be close to the embedding translation of object o_i by relation r , namely $o_i \simeq o_j + r$. Then, extensions have been proposed considering, e.g., different object representations according to the semantic relation type (TransH) [33] or a dynamic mapping between objects and relations constrained by their diversity (TransD) [14].

Moreover, knowledge resources have been used to enhance the distributed representation of words for representing their underlying concepts [9, 18, 37, 38]. For instance, Faruqui et al. [9] propose a “retrofitting” technique consisting in a leveraging of lexicon-derived relational information, namely adjacent words of concepts, to refine their associated word embeddings. Other work [37, 38] proposes an end-to-end oriented approach that rather adjusts the objective function of the neural language model by either leveraging the relational and categorical knowledge to learn a higher quality word embeddings (RC-NET model) [37] or extending the CBOW model [38] with a function based on prior relational knowledge issued from an external resource.

2.2 On using Knowledge Resources in IR.

Both general/specific linguistic bases (e.g., WordNet or UMLS respectively) and large-scale knowledge graphs (e.g., Freebase) represent external resources that offer valuable information about word semantics through objects (e.g., words, entities, or concepts) and their associated relations (e.g., “is-a”, “part-of”). Based on the use of such resources, a first line of work in IR aims at increasing the likelihood of term overlap between queries and documents through query expansion [25, 36] or document expansion [1]. Among models expanding queries, Xiong et al. [36] propose two algorithms relying on the category of terms in FreeBase. While the unsupervised approach estimates the similarity between the category distribution of terms in documents and queries, the supervised approach exploits the ground truth to estimate the influence of terms. Authors in [25] propose a query expansion technique using terms extracted from multiple sources of information. For each query term, candidate expansion terms in top retrieved documents are ranked by combining their importance in pseudo-relevant documents and their semantic similarity based on their definition in WordNet. Differently, Agirre et al. [1] propose a document expansion technique based on the use of a random walk algorithm identifying from WordNet the most related concepts. The second line of work leverages relations modeled in knowledge resources at the document ranking level [35]. For instance, authors in [35] propose a learning-to-rank algorithm based on objects of knowledge resources that are related to a given pair of query-document.

2.3 On using Deep Neural Networks in IR.

Recently, a large amount of work has shown that deep learning approaches are highly efficient in several IR tasks (e.g., text matching [12], question-answering [4]). A first category of work uses neural models for IR tasks [2, 23, 39] to integrate embeddings in IR

¹<https://www.nlm.nih.gov/mesh/>

²<http://wordnet.princeton.edu>

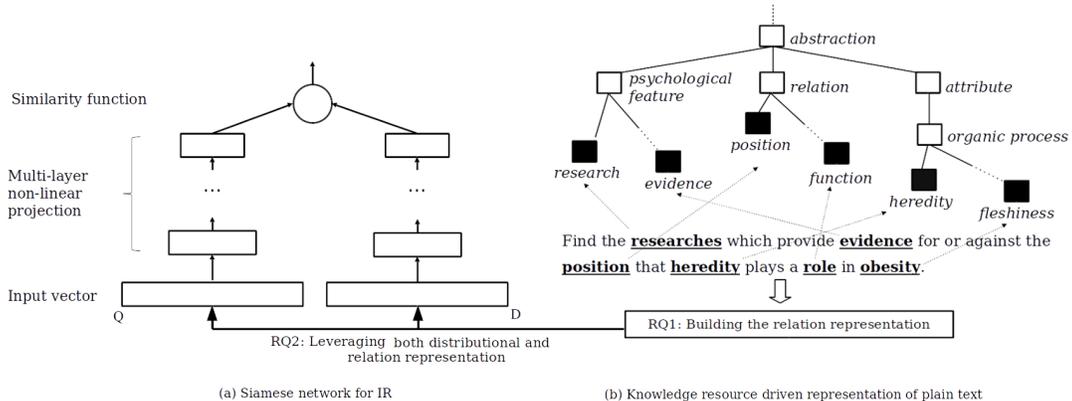


Figure 1: The general issue of our contribution: matching query and documents through a deep neural network by leveraging objects and relations expressed knowledge resources.

relevance functions. The second category of work, closer to our contribution, consists in end-to-end scoring models that learn the relevance of document-query pairs via latent semantic features [11, 12]. For instance, the Deep Semantic Structured Model (DSSM) [12] performs well on web search tasks. This model applies a siamese deep feed-forward network on document and query representations obtained by a word hashing method. The network aims at learning their latent representations and then measuring their relevance score using a cosine similarity. As an extension of the DSSM, Shen et al. [31] propose to use a convolutional-pooling structure, called Convolutional Latent Semantic Model (CLSM). In the same mind, Severyn and Moschitti [30] apply a convolution on the input layer to learn the optimal representation of short text pairs as well as the similarity function. However, these model parameters are hard to learn, which leads authors to only consider the matching between query-title pairs. To bypass this limitation, another line of work [10, 19, 26] rather aims at building a local interaction map between inputs, and then uses deep neural networks to learn hierarchical interaction patterns. For instance, the DeepMatch model [19] integrates a topic model into a fully connected deep network based on the word interaction matrix while the MatchPyramid model [26] applies a convolution to an interaction matrix estimated on the basis of word representation. Guided by the intuition that interaction matrix is more appropriate for the global matching and lacks of the term importance consideration, authors in [10] propose to model local interactions of terms through a histogram estimating the matching occurrences of query-document terms.

3 MOTIVATION

The literature review highlights that: 1) plain text and knowledge resources are complementary for both learning distributional representations and enhancing IR effectiveness [1, 25, 36], and that 2) neural approaches in IR have a great potential for ad-hoc search but could still be improved to compete with traditional IR models [10]. In this contribution, we address the problem of bridging the semantic gap in IR by leveraging both deep learning approaches [12, 30] and valid knowledge expressed in knowledge resources [25, 36]. In contrast to previous work on the semantic representation of

objects and relations leveraging knowledge resources [5, 9, 33, 37], our main concern is to build a semantic representation of documents that simultaneously takes into consideration objects and their pairwise relations expressed in a knowledge resource. As shown in Figure 1, we investigate the potential of siamese neural architectures, such as DSSM [12], and propose a model able to match queries with full texts, instead of titles as done in [12, 30], by jointly leveraging distributional and relational semantics of texts.

Accordingly, we address here two main research questions:

- **RQ1:** How to model the relational semantics of texts at the raw data level by jointly leveraging objects and their relations expressed in knowledge resources?
- **RQ2:** How to jointly learn the query-document relevance function and a representation of text combining relational and distributional semantics?

4 DSRIM: DEEP SEMANTIC RESOURCE INFERENCE MODEL

We detail here our DSRIM model aiming at tackling the semantic gap between documents and queries through a deep neural approach leveraging knowledge resources. More particularly, our contribution relies on 1) the modeling of a knowledge resource-driven representation of texts at the raw data level taking into account the semantics of objects identified in a text and their underlying relations expressed in a knowledge resource, and 2) a siamese neural network architecture aiming at learning the text latent representation and the ranking function on the basis of a vector modeling both the plain text and its relational semantics expressed through knowledge resources. This two-branch architecture is used in several contributions in IR [11, 12, 30] but, in contrast to work relying on raw-text features, we use a semantic-based input. Below, we present our semantic representation of texts based on knowledge resources and then present the network architecture.

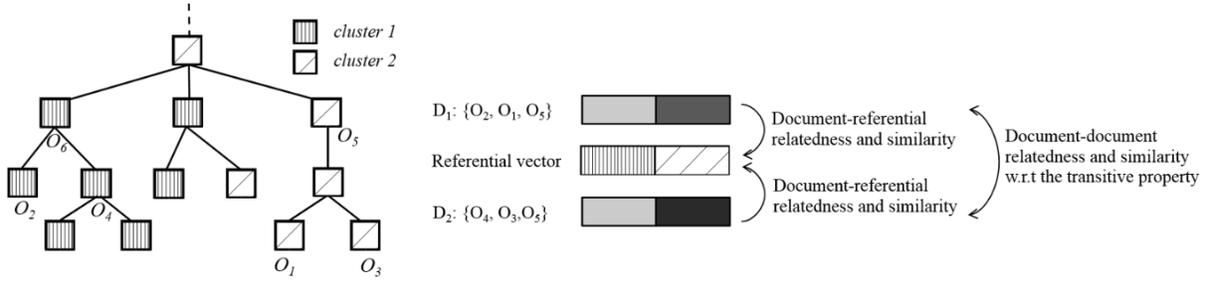


Figure 2: Intuition of the transitive property in knowledge resource-driven representation of documents

4.1 Knowledge Resource-driven Representation Vectors

Our aim here is to model a document/query representation that conveys their semantics with respect to a knowledge resource. The premise of this representation relies on the following assumptions: (A1) a text is a bag of identified objects from a knowledge resource, and (A2) semantically similar texts are deemed to entail similar and related objects.

Formally, a knowledge resource is built upon a relational graph $G = (V, E)$ where V is a node set and E is a edge set. Each node $v_i = \langle o_i, desc_i, \rangle$ includes an object o_i (e.g., word, entity) and its textual label $desc_i$ (e.g., preferred entry). Each object o_i is associated to a distributional representation x_i^d (e.g., its *ParagraphVector* [2] obtained on the basis of its textual labels $desc_i$). Each edge $e_{i,i'}$ expresses a semantic relation between objects o_i and $o_{i'}$. We suppose that given the set O of objects in the knowledge resource G , we can identify, for each text T , a set $O(T) \subset O$ of objects o .

While assumption A1 is easy to formalize through a binary vector modeling objects $o_i \in V$ or a vector combining their distributional representation x_i^d , it does not allow to fulfill assumption A2. To cope with this issue, the perspective of a vector representing object-object pairs could be a good option to simultaneously capture: 1) the objects belonging to a text, 2) their similarity as well as their relatedness. However, the large number of potential pairwise objects, or more precisely object-to-object relations, in a knowledge resource would lead to a high dimensional and sparse vector. To face this issue, we propose the *relation mapping method*, that: 1) similarly to the word hashing method [12], aims at reducing the dimensionality and the sparsity of the vector representation to make it scalable, and 2) allows building representations of both objects belonging to text T and their relations according to assumption A2. We describe below our approach for achieving these two sub-goals.

- Sub-goal 1) *Text representation vector space*: A naive approach consists in considering objects from the knowledge resource as unit vectors of a $|V|$ -dimensional space. Even if the number of objects in the resource is significantly lower than the number of object-to-object relations, the scalability of the underlying framework remains questionable. To fit with sub-goal 1) and lower the dimensionality of the vectorial representation space, we rather consider clusters of objects as representative of each dimension

of the vectorial space. Assuming that object-to-object relations might express topical relatedness between objects, we propose to build k topical clusters c_j of objects $o_i \in O$ assumed to be mutually independent. The latter refers to the referential $\mathcal{R} = \{c_1, \dots, c_k\}$ of the knowledge resource. In practice, we use the k-means clustering algorithm on the topical representation of objects, where the number of topical clusters k would be experimentally tuned (see Section 6.1). Thus, we consider a k -dimensional space, in which k is the number of topical clusters of objects.

- Sub-goal 2) *Knowledge resource-driven text representation*: The representation x^{KR} of text T (document or query) is a k -dimensional vector $x^{KR} = (x_1^{KR}, x_2^{KR}, \dots, x_k^{KR})$. To fulfill sub-goal 2, our intuition is that two documents are likely to be similar if they mention objects that are gathered around the same topical clusters. Naturally, the degree of similarity between those documents would depend on the average relatedness and similarity of their objects with each object in the topical clusters c_j of the referential \mathcal{R} . This refers to as a transitive property, illustrated in Figure 2. Each document D_1 and D_2 is modeled through a 2-dimensional vector in which each element represents a topical cluster. The gray levels in the document representation express the relatedness and similarity degree of document objects with respect to the topical clusters. Although documents D_1 and D_2 are not characterized by the same objects, they are as close to the referential, and accordingly, have similar representations.

We compute the components x_j^{KR} with respect to assumption A2. These values are expressed as a combination of the importance w_j^T of the topical cluster c_j according to text T and the relatedness $S_{relat}(c_j, O(T))$ of objects $O(T)$ belonging to text T with respect to topical cluster c_j , given by the following formula:

$$x_j^{KR} = w_j^T * S_{relat}(c_j, O(T)) \quad (1)$$

4.1.1 *Topical cluster importance score*. The importance score w_j^T of topical cluster c_j expresses to what extent the set $O(T)$ of objects belonging to text T are topically similar to objects belonging to topical cluster c_j . Intuitively, the degree of topical cluster contribution allows discriminating documents according to their degree of topical matching with those clusters. Accordingly, the more topically similar the objects mentioned in the representations of texts T and T' with respect to the topical clusters, the more similar texts T and T' .

Assuming that objects belonging to a text represent a topical cluster, we rely on previous work dealing with clustering similarity [15] suggesting to estimate the similarity between two sets of objects by aggregating similarities between objects of these two different sets. More formally, the topical cluster importance score between topical cluster c_j and object set $O(T)$ is estimated as:

$$w_j^T = \text{Agg_Function}_{(o_m, o_n) \in O(T) \times c_j} \text{sim}_t(o_m, o_n) \quad (2)$$

where *Agg_Function* expresses an aggregation function (we consider here the maximum to capture the best topical similarity between objects); sim_t estimates the topical similarity between vector representations of objects (here, the cosine similarity between the vectorial representations of object textual descriptions).

4.1.2 Topical cluster-text relatedness score. The topical cluster-text relatedness score $S_{\text{relat}}(c_j, O(T))$ measures to what extent objects $o_i \in O(T)$ belonging to text T are related to those of topical cluster c_j . Our intuition is that if the objects mentioned in texts T and T' are related to the representative of the same topical clusters, texts T and T' are more likely to be similar. Having in mind that state-of-the-art relatedness measures [28] rely on the computation of paths between objects, a scalable way allowing to measure this score is to consider the relatedness of objects $O(T)$ with respect to a representative object $R(c_j)$ of topical cluster c_j (e.g., the most frequent object in the collection among objects belonging to topical cluster c_j). The impact of the method used for identifying the representative is experimentally investigated (see Section 6.1). More formally, given a representative object $R(c_j)$ of topical cluster c_j , the topical cluster-text relatedness $S_{\text{relat}}(c_j, O(T))$ estimates the path length between object $R(c_j)$ and the object set $O(T)$:

$$S_{\text{relat}}(c_j, O(T)) = \sum_{o_m \in O(T)} \log(1 + \text{sim}_r(R(c_j), o_m)) \cdot \frac{\text{avg_no}}{|O(T)|} \quad (3)$$

where o_m is an object of the object set $O(T)$ characterizing text T . sim_r is a relatedness measure between objects (eg. the Leacock measure [17]); *avg_no* is the average number of objects belonging to documents in the collection. We introduce a normalization factor $\frac{\text{avg_no}}{|O(T)|}$ to avoid bias due to possible significant differences in text lengths in terms of the number of objects.

4.2 Model Architecture

4.2.1 Input. Based on previous findings in semantic IR highlighting that the combination of evidence from both words and concepts is effective [36], each text T (whether extracted from a document or a query) is characterized by an input vector $x_{\text{input}} = (x^t, x^{KR})$ modeled as a vector composed of two parts:

- *Plain text representation* x^t . This feature represents words of full text T . Based on previous findings highlighting the effectiveness of distributed semantic representations (in contrast to sparse word count representations [30]) to tackle the issue of large vocabulary, we estimate the low-dimensional semantic vector by using the *ParagraphVector* model [2].

- *Knowledge resource-driven representation* x^{KR} . This feature expresses the objects belonging to text T and their semantic relations expressed in the knowledge resource. This representation is built upon the *relation mapping* method (see Section 4.1).

4.2.2 Learning the latent representation. For each sub-network branch, the input vector x_{input} of text T is projected into a latent space by means of L hidden layers l_i ($i = 1, \dots, L$) so as to obtain a latent semantic vector y combining the distributional and relational semantics of text T . Each hidden layer l_i and the latent semantic vector y are respectively obtained by the following non-linear transformations:

$$\begin{aligned} l_0 &= x_{\text{input}} \\ l_i &= f(W_{i-1} \cdot l_{i-1} + b_{i-1}) \quad i = 1, \dots, L \\ y &= f(W_L \cdot l_L + b_L) \end{aligned} \quad (4)$$

where W_i and b_i are respectively the weight matrix and bias term of the i^{th} layer. The activation function $f(x)$ performs a non-linear transformation, namely the ReLU: $f(x) = \max(0, x)$. The use of the ReLU function is motivated by the fact that it does not saturate to 1 when x is high in contrast to the hyperbolic tangent [12], avoiding to face to the gradient vanishing problem.

After obtaining the latent semantic vectors y_D and y_Q of document D and query Q through the non-linear transformations of hidden layers, the document-query cosine similarity score $R(D|Q)$ is estimated between vectors y_D and y_Q .

4.2.3 Loss function. Since the ad-hoc retrieval task refers to a ranking problem, we optimize the parameters of the neural network using a pairwise ranking loss based on the distance Δ of similarity between relevant document-query pairs, noted (Q, D^+) , and irrelevant document-query pairs, noted (Q, D_p^-) . Unlike [12], it worth mentioning that we use the hinge loss function, more adapted for learning-to-rank tasks [6]. To do so, we build a sample of document-query pairs in which we oppose, for the same query Q , one relevant document D^+ for n irrelevant documents D_p^- , $p = [1..n]$, as suggested in [12]. The difference Δ between the similarity of the relevant pair (Q, D^+) and the irrelevant ones (Q, D_p^-) is defined as:

$$\Delta = \left[\text{sim}(Q, D^+) - \sum_{p=1}^n \text{sim}(Q, D_p^-) \right] \quad (5)$$

where $\text{sim}(\bullet, \bullet)$ is the output of the neural network.

Then, the DSRIM network is trained to maximize the similarity distance Δ using the hinge loss function L :

$$L = \max(0, \alpha - \Delta) \quad (6)$$

where α is the margin of L , depending on the Δ range.

5 EVALUATION PROTOCOL

We present here the experimental evaluation which, respectively to research questions RQ1 and RQ2 (Section 3), aims at assessing 1) the quality of our knowledge resource-driven representation of texts and 2) the effectiveness of our DSRIM model.

5.1 Datasets

We consider two datasets (statistics are presented in Table 1):

- The GOV2³ dataset gathering .gov sites used in the TREC Terabyte campaign. We use topics from the 2004, 2005, and 2006 campaigns and the narrative part of each topic as a query.

³http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm

Table 1: Statistics of the GOV2 and the PMC datasets

	GOV2	PMC
# Documents	25,000,000	733,138
Average length of documents (#words)	1132.8	477.1
# Queries	150	60
# Relevant pairs	25,100	8,346

- The PMC OpenAccess⁴ dataset containing biomedical full-texts from PubMed used in the TREC-CDS campaign. The summaries of topics of the 2014 and 2015 evaluation campaigns are used as queries.

To learn the text semantics, we use external knowledge resources matching each dataset application domain. For GOV2, we consider WordNet⁵ which is an English lexical database including about 117,000 synsets (groups of words associated with the same concept). These synsets are connected by 6 semantic relations. For instance, the most common one is the “IS-A” one (hyponymy or hyperonymy relation) which we exploit in our experiments. For the PMC dataset, we use the 2015-Mesh version⁶, produced by the National Library of Medicine. This resource includes over 27,000 concepts, also called descriptors, organized in 16 categories, hierarchically structured from the most general to the most specific in up to 13 levels.

5.2 Implementation Details and Evaluation Methodology

To build the input layer, we pre-train a *ParagraphVector* model on the plain text corpus for learning vector x^t . The vectors are sized to 100, which is consistent with previous results outlining that low-dimensional paragraph vector models are able to capture complex topic structures [2]. The concepts used for building our knowledge resource-driven representation are extracted using appropriate tools, namely *SenseRelate* [27] for the GOV2 dataset and *Cxtractor*⁷ relying on *MaxMatcher* [41] for the PMC dataset. Concerning our model architecture, we set the number of hidden layers to 2 with a hidden vector size equals to 64 leading to an output layer of 32 nodes. Similarly to [12], the number n of irrelevant document-query pairs opposed to a relevant one is 4 (Equation 5). Relevant/irrelevant document-query sets are randomly extracted from each dataset ground truth, supplying graded relevance judgments from 0 to 2 (relevance criteria: 1 and 2).

To train our model parameters, we apply the 5-fold cross-validation method. The topics in each dataset are divided into 5 folds. For each fold retained as the test set for model evaluation, the other 4 folds are used to train and validate the model. The final retrieval results are averaged over the test results on 5 folds. The model is optimized using a 5-sample mini-batch stochastic gradient descent (SGD) regularized with a dropout equals to 0.3. Our model generally converges after 50 epochs over the training dataset.

⁴<https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/>

⁵<http://wordnet.princeton.edu>

⁶<https://www.nlm.nih.gov/mesh/>

⁷<https://sourceforge.net/projects/cxtractor/>

For evaluating the ranking performance of our model and the different baselines, we perform a re-ranking [3] which is carried out over the top 2,000 documents retrieved by the BM25 model. Final results are estimated using the top 1000 documents of each re-ranking model according to the MAP metric.

5.3 Parameter Settings

According to our first experimental goal dealing with the quality of the knowledge resource-driven representation x^{KR} , we adopt different settings related to two parameters: 1) the number of topical clusters: we set the number k of topical clusters to $k \in \{100, 200\}$; 2) the choice of the representative object $\mathcal{R}(c_j)$ within each topical cluster: we use three strategies: *idf_{min}*, namely the most frequent object; *idf_{max}*, the less frequent one; and *centroid*, the closest object to the centroid, that have been evaluated with respect to a naive baseline. The latter consists in selecting the top k frequent objects in the document collection as the representative objects, where $k \in \{100, 200\}$. Combining these settings leads to eight possible configurations of our knowledge resource-driven representation.

5.4 Baselines

Our second experimental goal refers to the effectiveness evaluation of our model. To measure the impact of the different evidence sources taken into consideration for representing texts (query and documents), three scenarios built upon our model are presented:

- *DSRIM^{p2v}*: Our proposed neural model based on an input representation of texts restricted to the plain text, namely x^t .
- *DSRIM^{kr}*: Our proposed neural model based on our knowledge resource-driven representation of text, namely x^{KR} .
- *DSRIM^{kr+p2v}*: Our proposed neural model based on an enhanced representation of texts combining plain text representation x^t and our knowledge resource-driven representation x^{KR} .

To evaluate our model effectiveness, we use three types of baselines: exact term matching models (BM25, LM-DI) to highlight the impact of both leveraging relation semantics and deep learning approaches, latent semantic based matching models (QE, LM-LDA) to outline the impact of a deep neural approach guided by knowledge resource for learning text semantics, and deep neural semantic matching models (DSSM, CLSM) based on a siamese:

- **BM25**: The well-known probabilistic model (*BM25*).
- **LM-DI**: The language model based on Dirichlet smoothing which is another effective retrieval model with exact term matching [40].
- **LM-QE**: A language model applying a concept-based query expansion technique [25] in which candidate terms are ranked based on their similarity with descriptions in the knowledge resource. Default parameters mentioned in the paper are used.
- **LM-LDA**: The LM-LDA is a latent model using the language modeling framework [34]. To perform a fair comparison with our model, we set the number of topics equal to the size of the output vector y in our model, namely 32.
- **DSSM**: The state-of-the-art deep structured semantic matching model [12]. We adopt the publicly released code⁸ with default parameter values. We also evaluate the DSSM on full-text documents.
- **CLSM**: The DSSM extension model in which the feed-forward

⁸<https://www.microsoft.com/en-us/research/project/dssm/>

Table 2: Cosine similarities of the knowledge resource-driven representation on most similar (Top_10) and less similar (Less_10) documents, averaged on 100 random pivotal documents. diff: difference between Top_10 and Less_10

			GOV2			PMC		
	#Clusters k	Repres. obj. $R(c_i)$	Top_10	Less_10	diff	Top_10	Less_10	diff
Clustering	#Cluster 100	idf_max	0.7490	0.5776	0.1714	0.5455	0.3035	0.2420
		centroid	0.7411	0.5693	0.1719	0.4807	0.2862	0.1945
		idf_min	0.7018	0.5501	0.1518	0.4975	0.2717	0.2259
	#Cluster 200	idf_max	0.7595	0.5814	0.1781	0.6359	0.3885	0.2475
		centroid	0.7344	0.5536	0.1808	0.6464	0.3842	0.2621
		idf_min	0.7645	0.5660	0.1985	0.6485	0.4234	0.2251
Top frequent concepts (naive baseline)	Top 200		0.9034	0.9013	0.0021	0.9861	0.9616	0.0245
	Top 100		0.9123	0.9049	0.0074	0.9817	0.9572	0.0245

neural network is replaced by a convolutional network for better capturing fine-grained contextual structures [31]. We also apply the publicly released CLSM code⁸ on full-text documents and use the default parameter values.

6 RESULTS

6.1 Analyzing the Semantic Representation of Documents

In this section, we propose to analyze our knowledge resource-driven representation through a twofold objective: 1) identifying the optimal parameter setting of the vectorial representation and 2) assessing the validity of the built document vectors x^{KR} .

To perform our analysis, we assess the vectorial representation quality based on the intuition that semantically similar texts, modeled as bags of concepts, should have similar vectorial representations built following our approach; such representations should also discard non-similar documents [16, 21]. In practice, given a randomly selected document (called a “pivotal document”), a good vectorial representation should 1) ensure that the distance between the pivotal document and each other document of the collection is non-uniform, and 2) maximize the distance between its most similar documents and its less similar ones. To this end: 1) we first identify for each given pivotal document, the set \mathcal{D}_+^p of its 10 most semantically similar documents and the set \mathcal{D}_-^p of the 10 less semantically similar documents over the whole dataset using a concept-oriented metric proposed in [7], called in the remaining the *Corley* measure; and 2) then we compute the average cosine similarity of the representations of the pivotal documents with the sets \mathcal{D}_+^p and \mathcal{D}_-^p . Table 2 presents the comparative results of the eight settings for 100 randomly selected pivotal documents and suggests the following statements:

- Regarding the method used for defining the vectorial representation space (See sub-goal 1; Section 4.1), we can see that our proposed approach for identifying the referential based on the object clustering is more effective than the naive baseline based on the top frequent concepts. Indeed, the similarity differences of both document sets are very small (< 0.02 for both datasets vs. higher than 0.5 for our clustering approach) and cosine values between the pivotal documents and the most/less similar documents are very

high (> 0.9 for both datasets). In contrast, we outline that cosine values for our clustering approach seem to be more intuitive, with an average cosine for the GOV2 dataset higher than 0.6 for the most similar documents and lower than 0.6 for the less similar ones (respectively 0.5 for the PMC dataset). These statements suggest that our referential building approach based on topical clustering seems reasonable. Moreover, the difference in terms of cosine value range between both datasets (higher for the GOV2 dataset) conjectures that representing texts using objects and relations expressed in a knowledge resource seems to be more difficult for the PMC dataset. This could be explained by the fact that this dataset focuses on a particular application domain (namely, the medical) that might implies a more technical vocabulary, in contrast to the GOV2 dataset, which remains on generic word senses. Moreover, we can see that the dimension used for the vectorial representation k impacts the similarity degree between documents. Indeed, the average similarities between pivotal documents and the set of top similar ones are more important for a higher number of clusters (e.g., up to 0.6485 for $k = 200$ vs. 0.5455 for $k = 100$ for the PMC dataset). Also, this setting allows obtaining higher differences between the most vs. less similar documents (with at least 0.2251 vs. 0.1945 for respectively $k = 200$ and $k = 100$ for the PMC dataset, and 0.1781 vs. 0.1518 for the GOV2 dataset). These results highlight the importance of achieving a reasonable ratio between the knowledge resource size (in terms of the number of object-object-relations) and the number of representative clusters of objects to better capture the semantic representation of documents.

- Focusing on the methods used for the knowledge resource-driven representation (See sub-goal2; Section 4.1) and more particularly, the one used for choosing the topical cluster representative (See Section 4.1.2), we can notice that, for $k = 200$, the best scenario consists in selecting the closest object to the cluster centroid (*centroid*) for the PMC dataset while these are no significant differences between the three methods for the GOV2 dataset. Given that the centroid method is more intuitive with the assumptions used for building the referential, we retain the setting with 200 topical clusters and the centroid method for extracting the representative object.

Table 3: Effectiveness comparison of baselines and DSRIM on GOV2 and PMC collections. % Chg: Significant improvement/degradation of $DSRIM^{kr+p2v}$ w.r.t each baseline is indicated (+/-). p -value: Significance t-test: * : $0.01 < \alpha \leq 0.05$, ** : $0.001 < \alpha \leq 0.01$, * : $\alpha \leq 0.001$**

		GOV2			PMC		
Model Type	Model	MAP	%change	p-value	MAP	%change	p-value
Exact	BM25	0.1777	+4.84	0.6691	0.0348	-1.15	0.9628
Matching	LM-DI	0.1584	+17.61	0.1644	0.0379	-9.23	0.7109
Semantic	LM-QE	0.0738	+152.44	0.0001 ***	0.0106	+224.53	0.0008 ***
Matching	LM-LDA	0.0966	+92.86	0.0001 ***	0.0185	+85.95	0.0323 *
Deep	DSSM	0.0418	+345.69	0.0001 ***	0.0095	+262.11	0.0008 ***
Matching	CLSM	0.0365	+410.41	0.0001 ***	0.0069	+398.55	0.0001 ***
Our approach	$DSRIM^{p2v}$	0.1115	+67.09	0.0001 ***	0.0183	+87.98	0.0460 *
	$DSRIM^{kr}$	0.1801	+3.44	0.7461	0.0307	+12.05	0.6829
	$DSRIM^{kr+p2v}$	0.1863			0.0344		

6.2 Measuring the Model Effectiveness

We present here the performance of our model on both datasets GOV2 and PMC. Table 3 shows a summary of effectiveness values in terms of MAP for our model and the different baselines.

Comparing different configurations of our approach, namely $DSRIM^{p2v}$, $DSRIM^{kr}$, and $DSRIM^{kr+p2v}$, we can see that the DSRIM model applied only on our knowledge resource-driven representation x^{KR} provides better performance according to the MAP metric than the one with only the plain text-based representation x^t (e.g., respectively 0.0307 and 0.0183 for the PMC dataset). This result reinforces the intuition claimed in recent work dealing with the use of text representations based on local interactions of terms and/or non-learned features [10]. Moreover, when combining the distributional and the relational semantics through the $DSRIM^{kr+p2v}$ model, we could see that the MAP value slightly increases, with for instance a significant improvement of +67.09% and +87.98% for the GOV2 and PMC datasets respectively with respect to $DSRIM^{p2v}$. This opens interesting perspectives in the combination of those word-sense approaches as we claim in this paper.

With this in mind, we comment the baseline comparison with respect to the $DSRIM^{kr+p2v}$ model. From a general point of view, we can see, on the one hand, that exact matching models are non-significantly different from our proposed model, with a particular attention to the GOV2 dataset with small improvements with respect to BM25 (+4.84%) and LM-DI (+17.61%). On the other hand, our approach overpasses semantic and deep matching models with significant improvements. For instance, our model reports significant better results for the GOV2 dataset according to the MAP compared with the LM-QE, LM-LDA, DSSM, and CLSM baseline models for which our model obtains a MAP value up to +410.41% of improvement rate. Those observations are similar for both datasets, highlighting the fact that our model is effective for leveraging general (WordNet) as well as domain-oriented (MeSH) knowledge resources. More particularly, we can formulate the following statements:

- The BM25 and the language models are well-known as strong baselines in IR which are difficult to outperform with deep matching models learned with small training datasets that do not allow to generalize the task. It is worth noting that, in contrast to most

previous neural approaches [12, 30, 31] that rank short documents (titles) and use large-scale real collection for training their model, we rather experiment our model on long full-text document collections (average length is 1132.8 words for the GOV2 and 477.1 for the PMC datasets). To get a better understanding of this result, we investigate to what extent the effectiveness of our model depends on the level of difficulty of queries. More particularly, we classify queries according to three levels of difficulty (“easy”, “medium”, “difficult”) using the k-means algorithm applied on the BM25 MAP values. Statistics of the obtained classes are presented in Table 4. We can outline that, for the PMC dataset, difficult queries significantly include more terms and more objects than easy and medium ones. However, there is no significant differences between the different query types with respect to the number of terms and objects for the GOV2 dataset. Focusing on the retrieval effectiveness, it can be seen that $DSRIM^{kr+p2v}$ improvements according to BM25 are both positive and significant for difficult queries for both GOV2 and PMC datasets. Moreover, it is worth mentioning that the improvement rates for difficult queries (+63.60% for the PMC dataset) are significantly different from the ones for medium and easy queries (respectively -25.78% and -0.22% for the PMC dataset, with no significant improvement difference between easy and medium queries, $p > 0.5$). Interestingly, combining the improvement rates and the number of objects for medium queries of the GOV2 dataset, we can see that the significant effectiveness decrease of our model (-5.15%) could be explained by the lowest number of objects associated to this query set. These results highlight that leveraging the relational semantics through our knowledge resource-driven representation is more effective for solving difficult queries. This

Table 4: Statistics on queries w.r.t their difficulty level

		Difficulty level	#Words	#Objects	%Change
GOV2	Easy		22.95	12.11	-16.60%
	Medium		20.79	11.79	-5.15%*
	Difficult		22.15	12.14	+87.15%***
PMC	Easy		13	5.4	-0.22%
	Medium		16.68	5.36	-25.78%*
	Difficult		18.5	6.3	+63.60%*

Table 5: Average similarity of relevant document-query pairs at the I/O layers

	GOV2		PMC	
	Input	Output	Input	Output
DSSM	0.1482	0.3955	0.1049	0.1111
DSRIM	0.1988	0.7386	0.3659	0.6283

is coherent since those queries are generally characterized by a high number of words and extracted objects. Accordingly, we can reasonably argue that our model is particularly devoted to lower the semantic gap between word-based and concept-based representations of documents and queries which probably favors the discrimination between relevant and irrelevant documents.

- The LM-QE baseline performs a knowledge resource-based query expansion. Since the DSRIM outperforms the LM-QE model, we can suggest that the semantic based representations of documents and queries which are learned starting from the input built upon the relation mapping method, is more effective than the expanded queries with relevant object descriptors.

- The LDA-LM model is based on a probabilistic generative model able to identify relevant topics. Our model generally outperforms this baseline with a significant improvement of 89.95% for the MAP metric on the PMC dataset. This is consistent with previous work [12], highlighting the effectiveness of deep latent representations of texts in comparison to those obtained by generative models.

- In the category of neural IR models, our model outperforms the DSSM and the CLSM models (with a MAP reaching 0.0418 and 0.0095 for both datasets respectively). These results suggest that the integration of relational as well as the distributional semantics at the document level (rather than the word level) into the input representation allows enhancing the learning of the deep neural matching model. Interestingly, the convolutional CLSM model initially overpassing the DSSM in [31] through experiments carried out on a large-scale real-world data, is less effective than the DSSM. One explanation might be that it is trained using TREC collections characterized by a limited number of queries (as also shown in [10]). This observation combined with the comparison of our model with the BM25 baseline gives rise to research opportunities in neural IR in terms of representation learning on small training datasets by introducing for instance distant supervision approaches.

In order to further investigate the impact of incorporating evidences issued from the external knowledge in a deep model, we report in Table 5 the measures of the cosine similarity between document-query vectors of input and output relevant pairs obtained using both DSSM and DSRIM. As can be seen from Table 5, although input similarities are of the same range for both datasets, the similarity improvement between input/output representations is more important for our model than for the DSSM model for both datasets: 166.88% for DSSM vs. 271.51% for DSRIM for the GOV2 dataset, 5.91% for DSSM vs. 71.71% for the PMC dataset. These results suggest that the use of evidence from relational semantics underlying queries and documents allows a better discrimination between relevant and irrelevant documents.

7 CONCLUSION

We propose the DSRIM model, a deep neural IR model combining the distributional and the relational semantics underlying texts. While the former is modeled using state-of-the-art work, namely the *ParagraphVector* algorithm, the latter is modeled through a knowledge resource-driven representation of texts aiming at jointly modeling objects and structured relations between objects from a knowledge resource. To tackle the issue faced by the high-dimensionality representation of pairwise object relations, we propose the relation mapping method based on the premise that similar documents entail similar and related concepts. Experimental evaluation on two TREC datasets, namely the GOV2 and the PMC Open Access, are performed. Results show that 1) our knowledge resource-driven representation allows to discriminate similar from non-similar texts, and 2) our model overpasses semantic-driven approaches as well as state-of-the-art neural IR models. In the near future, we plan to further the knowledge resource-driven representation by taking into account both the heterogeneity of objects and the heterogeneity of the relations between objects. Another interesting perspective would be to explore the usefulness of the relation mapping method for estimating local interactions of terms/objects between queries and documents, as [10, 19, 26].

ACKNOWLEDGEMENT

This research was supported by the French FUI research program SparkInData.

REFERENCES

- [1] Eneko Agirre, Xabier Arregi, and Arantxa Otegi. 2010. Document expansion based on WordNet for robust IR. In *ICCL*. 9–17.
- [2] Qingyao Ai, Liu Yang, Jiafeng Guo, and W. Bruce Croft. 2016. Analysis of the Paragraph Vector Model for Information Retrieval. In *ICTIR*. ACM, 133–142.
- [3] Qingyao Ai, Liu Yang, Jiafeng Guo, and W. Bruce Croft. 2016. Improving Language Estimation with the Paragraph Vector Model for Ad-hoc Retrieval. In *SIGIR*. ACM, 869–872.
- [4] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering with Subgraph Embeddings. In *EMNLP*. 615–620.
- [5] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.
- [6] Wei Chen, Tie yan Liu, Yanyan Lan, Zhi ming Ma, and Hang Li. 2009. Ranking Measures and Loss Functions in Learning to Rank. In *NIPS*. 315–323.
- [7] Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Workshop on empirical modeling of semantic equivalence and entailment*. ACL, 13–18.
- [8] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391–407.
- [9] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *NAACL*.
- [10] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM*. 55–64.
- [11] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*. 2042–2050.
- [12] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. 2333–2338.
- [13] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *ACL*. 95–105.
- [14] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *ACL*. 687–696.
- [15] Benjamin King. 1967. Step-Wise Clustering Procedures. *J. Amer. Statist. Assoc.* 62, 317 (1967), 86–101.
- [16] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*. 1188–1196.

- [17] Claudia Leacock and Martin Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database* 49, 2 (1998), 265–283.
- [18] Xiaojie Liu, Jian-Yun Nie, and Alessandro Sordoni. 2016. Constraining Word Embeddings by Prior Knowledge – Application to Medical Information Retrieval. In *AIRS*.
- [19] Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *NIPS*. 1367–1375.
- [20] Andrew McCallum. 2015. Embedded Representations of Lexical and Knowledge-Base Semantics. In *ICTIR*. ACM, 1–1.
- [21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781* (2013).
- [22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. 3111–3119.
- [23] Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. 2016. A dual embedding space model for document ranking. *arXiv:1602.01137* (2016).
- [24] Gia-Hung Nguyen, Lynda Tamine, Laure Soulier, and Nathalie Bricon-Souf. 2016. Toward a deep neural approach for knowledge-based IR. *arXiv preprint arXiv:1606.07211* (2016).
- [25] Dipasree Pal, Mandar Mitra, and Kalyankumar Datta. 2014. Improving query expansion using WordNet. *Journal of the Association for Information Science and Technology* 65, 12 (2014), 2469–2478.
- [26] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching As Image Recognition. In *AAAI*. AAAI Press, 2793–2799.
- [27] Ted Pedersen and Varada Kolhatkar. 2009. WordNet::SenseRelate::AllWords: A Broad Coverage Word Sense Tagger That Maximizes Semantic Relatedness. In *NAACL-Demonstrations*. 17–20.
- [28] Ted Pedersen, Serguei V.S. Pakhomov, Siddharth Patwardhan, and Christopher G. Chute. 2007. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics* 40, 3 (2007), 288 – 299.
- [29] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.
- [30] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*. 373–382.
- [31] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM*. 101–110.
- [32] Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. In *SIGIR*. ACM, 363–372.
- [33] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*. 1112–1119.
- [34] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *SIGIR*. ACM, 178–185.
- [35] Chenyan Xiong and Jamie Callan. 2015. ESDRank: Connecting Query and Documents Through External Semi-Structured Data. In *CIKM*. 951–960.
- [36] Chenyan Xiong and Jamie Callan. 2015. Query expansion with Freebase. In *ICTIR*. ACM, 111–120.
- [37] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. Rc-net: A general framework for incorporating knowledge into word representations. In *CIKM*. 1219–1228.
- [38] Mo Yu and Mark Dredze. 2014. Improving Lexical Embeddings with Semantic Knowledge. In *ACL*. 545–550.
- [39] Hamed Zamani and W. Bruce Croft. 2016. Estimating Embedding Vectors for Queries. In *ICTIR*. ACM, 123–132.
- [40] Chengxiang Zhai and John Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *SIGIR*. 334–342.
- [41] Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. 2006. MaxMatcher: Biological Concept Extraction Using Approximate Dictionary Lookup. In *PRICAI*. Springer.