Knowl Inf Syst (2019) 58:551–583
https://doi.org/10.1007/s10115-018-1190-1

REGULAR PAPER

Ad hoc retrieval via entity linking and semantic similarity

Faezeh Ensan[1] · **Weichang Du**[2]

Received: 22 March 2017 / Revised: 6 February 2018 / Accepted: 12 April 2018 /
Published online: 21 April 2018

Abstract Semantic search has emerged as a possible way for addressing the challenges of traditional keyword-based retrieval systems such as the *vocabulary gap* between the query and document spaces. In this paper, we propose a novel semantic retrieval framework that uses semantic entity linking systems for forming a graph representation of documents and queries, where nodes represent concepts extracted from documents and edges represent semantic relatedness between those concepts. The core of our proposed work is a semantic-enabled language model that estimates the probability of generating query concepts given values assigned to document concepts. The semantic retrieval framework also provides basis for interpolating keyword-based retrieval systems with the semantic-enabled language model. We conduct comprehensive experiments over several Trec document collections and analyze the performance of different configurations of the framework across multiple retrieval measures. Our experimental results show that the proposed semantic retrieval model has a synergistic impact on the results obtained through the state-of-the-art keyword-based systems, and the consideration of semantic information can complement and enhance the performance of such retrieval models.

Keywords Semantic search · Ad hoc retrieval · Entity linking · Semantic relatedness · Language models

1 Introduction

Ad hoc keyword-based information retrieval (IR) systems, the core of many current search engines, find and rank relevant documents to user queries based on the frequent occurrence of query terms in the available documents. Keyword-based IR systems have limitations such as

✉ Faezeh Ensan
ensan@um.ac.ir

1 Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

2 Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada

the *vocabulary gap* and the *ambiguous keywords* problems, among others [9]. The vocabulary gap problem occurs in cases when users choose query terms that are different from the terms that are used in the documents for expressing the same meanings. Furthermore, the ambiguity problem refers to ambiguous keywords, which are words with more than one meaning in both the document and query spaces. For example, 'Java' may either refer to a programming language or an island in Indonesia. Retrieving documents that are about an incorrect sense of a word can decrease the precision of an IR system.

Semantic Information Retrieval (Semantic IR) is a step forward for tackling the existing challenges in keyword-based IR systems [37]. In Semantic IR, documents and queries are modeled as a set of meaningful concepts instead of a bag of words (BOW); hence, it is possible to search for the correct sense of concepts, to search based on the possible instantiation and subclass meta-information attached to the concepts, and to match queries and documents based on concepts.

In this paper, we introduce a semantic retrieval framework for improving the performance of keyword-based IR systems in ad hoc retrieval. The semantic retrieval framework uses semantic concepts in documents and queries and their relatedness[1] for the purpose of scoring and ranking. The field of automated semantic annotation of textual content for extracting concepts and entities and linking them to external knowledge bases [11,33], as well as computing semantic similarities between knowledge base entities [14,36], has been widely studied in the literature, and promising performance has been reported [6]. The retrieval framework presented in this paper can utilize any semantic annotation (entity linking) system. In fact, in the presented framework, any entity linking system can be used as a module that provides concept representation of queries and documents. Also, this frameworks can be configured to use any semantic relatedness technique for providing semantic relatedness between any given two concepts.

The core of the semantic retrieval framework is the semantic-enabled language model (SELM) [10], which provides the basis for ranking documents based on concepts and their relatedness. SELM models a document as an undirected graph where each node corresponds to a concept in the document and each edge represents a relatedness relationship between two concepts. In forming the graph, it is assumed that two concepts are related if there is an edge between their corresponding nodes in the graph and there is no dependency between two non-neighboring concepts. Based on this graph, SELM adopts a probabilistic reasoning model based on conditional random fields for calculating the conditional probability of a query concept (as the output label) given values assigned to document concepts (as input nodes). SELM uses the conditional probabilities for forming the language model.

The semantic retrieval framework proposed in this paper also provides a basis for interpolating semantic-based retrieval with other keyword-based retrieval system for producing a ranked list of results based on both semantic and syntactic features of documents and queries. In this paper, we thoroughly describe the main parts of the framework. We also explain our implementation method for SELM by expanding queries not with words or text but rather with a set of related concepts.

In our experiments, we evaluate the impact of SELM for improving existing retrieval systems. Our extensive experiments show that SELM is able to identify a distinct set of documents as relevant to user queries that were not retrieved by state-of-the-art retrieval models. In addition, we report that there are cases where the retrieval of keyword-based models is not included in SELM. Therefore, the integration of SELM and keyword-based models would collectively yield and retrieve a larger set of relevant results. We show in our

[1] While recognizing the differences, we use relatedness and similarity interchangeably in this paper.

experiments that the interpolation of keyword-based model with SELM will significantly enhance the performance of the-state-of-the-art models by identifying relevant documents that could not have been retrieved otherwise.

The major contributions of this paper are as follows:

- We propose a semantic retrieval framework for ad hoc retrieval that retrieves documents based on the degree of relatedness of the concepts within the query and document spaces. Our novel language model, SELM, for semantic retrieval estimates the score of a document for any given query based on a probabilistic reasoning model and conceptual representation of queries and documents.
- We comprehensively describe three configurations of the semantic retrieval framework, where it is configured to work with different semantic similarity estimation systems. We thoroughly evaluate the framework on its configurations over several benchmark document collections.
- Based on rigorous experiments on several benchmark collections and analyzing the performance of SELM in its different variations compared to the state-of-the-art, we show that there are many cases where the entity-based treatment of queries and documents can have synergistic impact on the results obtained through state-of-the-art keyword-based approaches. Based on these observations, we show that the consideration of semantic information obtained from entity linking on queries and documents can complement and enhance the performance of keyword-based retrieval models.

The rest of this paper is organized as follows: Sect. 2 introduces SELM and its main components. Section 3 presents SELM, the language model for semantic retrieval. Section 4 explains the interpolation with other retrieval systems. Section 5 studies implementing SELM by expanding queries with extra concepts. Section 6 introduces three variations of the semantic retrieval framework and details their implementations. Section 7 provides our experimental results over different benchmarks and with different variations introduced in Sect. 6. Section 8 discusses related work, and finally, Sect. 9 concludes the paper.

2 Semantic retrieval framework

In this section, we introduce our proposed semantic retrieval framework. Figure 1 depicts the main components of this framework and the way they fit together.

In this figure, documents and queries are first processed by a semantic annotation module, which is responsible for generating entity links for queries and documents. As we will show later in our experiments, the semantic annotation module can be implemented using any off-the-shelf entity linking tool that has the ability to perform spotting of key phrases in the text and linking them to concepts in a knowledge base such as DBpedia or Freebase.

In order to be able to not only search for the query terms but also concepts in the document space, our framework maintains an additional concept index. This approach contrasts some of the existing approaches in the literature [9] that index the related context of each concept in documents; hence, the concept indexer of this framework produces a considerably lower size index. For example, for a document containing the word 'Einstein,' where the annotation module finds a reference to the concept 'Albert Einstein' from DBpedia, the concept indexer will only index this concept and not the other related concepts such as 'Physicist' and 'The theory of relativity,' which can be later retrieved through the 'Albert Einstein' concept. Our approach also differs from the entity annotation and indexing method presented in [4] in which highly compressed data structures for spotting and disambiguating entity mentions

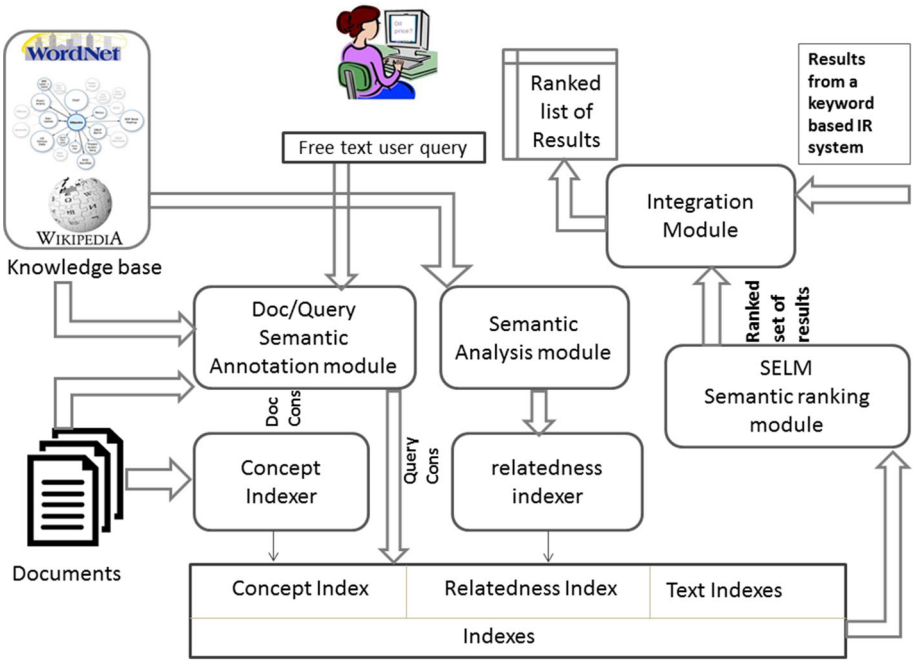


Fig. 1 Semantic retrieval framework

and indexing are presented. Contrary to [4], in our architecture, semantic annotations and concept indexing are designed as independent modules that despite dependencies can be replaced by any available alternatives. We refer the interested reader to our earlier work on building indices for semantic search [24].

Another core component of our work is the semantic analysis module which relies on semantic similarity metrics to calculate the degree of similarity or relatedness of two concepts. We additionally store the semantic relatedness values of concepts the first time they are computed so they can be retrieved through a simple lookup in the future.

Now, given an input user query, it is first processed by the semantic annotation module so that possible concepts within the query are identified. The concepts are then passed to the SELM semantic ranking module for producing a set of documents that are scored and ranked based on their similarity to the concepts observed in the input query.

The architecture depicted in Fig. 1 imposes a parallel hybridization design for integrating semantic retrieval and keyword-based IR systems. Based on this design, documents and queries are fed into any other IR system and the lists of results are interpolated with the result list of the semantic ranking module. The final results are produced by the integration module by interpolating semantic search with other possible keywork-based solutions.

3 SELM: semantic-enabled language model

In this section, we first provide preliminaries regarding fundamentals of language models and their scoring method. Next, we describe SELM and its main features through an illustrative example. Finally, we provide details of the scoring method in SELM.

3.1 Background

Language models have been widely studied and applied in different retrieval tasks due to their clearly defined statistical foundations and good empirical performance [40]. The *query likelihood* model is the basic method for using language models in information retrieval. Based on this model, for ranking document d given query q, $P(d|q)$ needs to be estimated, where the probability of a document given a query is interpreted as the relevance of the document to the query. Using Bayes rules, $P(d|q)$ can be calculated as follows:

$$P(d|q) = P(q|d)P(d)/P(q)$$

For the purpose of document ranking, $P(q)$ is ignored because it is identical for all documents. Also, $P(d)$ is often assumed to be uniform across all documents for the purpose of simplification,[2] so it can also be ignored in the ranking process. Consequently, documents are ranked based on $P(q|d)$, which is interpreted as the probability of generating query q using the language model derived from d. Here, the main idea is to estimate a language model θ_d for each document d and to rank documents based on the likelihood of generating the query using the estimated language models.

In other words, for ranking document d, the following scoring method is employed:

$$\text{Score}(d, q) = P(q|\theta_d)$$

where, θ_d, the language model estimated for document d, is a probability distribution over all possible query units, and $P(q|\theta_d)$ denotes the probability of query q according to distribution θ_d. Clearly, one of the important steps is the estimation method for finding θ_d. Keyword-based language modeling approaches primarily define the probability distribution based on the *exact* match of terms in the query and those in the documents as well as the collection of documents [32,40]. For example, the multinomial unigram language model, one of the most commonly used keyword-based methods, uses a multinomial distribution over words for estimating document language models. In contrast, our language model estimates the probability distribution based on semantic relatedness between concepts recognized in queries and documents.

3.2 The SELM model

Based on the language modeling approach to information retrieval, we assume that a query q is generated from a document d by the probabilistic model θ_d. Here we are interested in estimating $P(q|\theta_d)$ for the purpose of scoring and ranking d. SELM provides an estimation for $\theta_d = \{P(q_i|d)\}_{i \in \lfloor 1,|Q| \rfloor}$, where $P(q_i|d)$ is the probability of query q_i and Q is the set of all query units. We ensure that $\sum_{i \in \lfloor 1,|Q| \rfloor} P(q_i|d) = 1$. In estimating the probability distribution, we adopt an undirected graphical model for calculating the conditional probability of a set of target variables, given the set of observed variables. In the context of our model, concepts of the query are modeled as the target variables and concepts of the document are modeled as the set of observed variables.

Our undirected graphical model is similar to CRFs that have been previously applied to different information retrieval tasks. In work such as [29], CRFs are used for modeling sequential data. In these works, it is assumed that the output is a sequence of labels, and input variables and their dependencies form a chain. In [38,39], CRFs are used as a method for combining a diverse set of features. The challenging aspect of existing work is to efficiently

[2] $P(d)$ is used in some retrieval methods for modeling document-specific criteria such as authority.

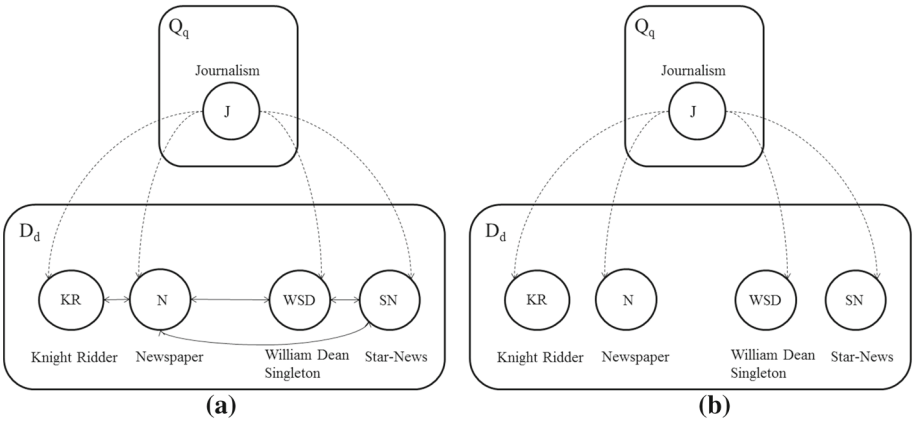


Fig. 2 Sample query and document relationship model. **a** Semantic relationships between concepts. **b** Relationships used by SELM for semantic ranking

learn appropriate weights for different feature functions based on the available training data. In this paper, we do not restrict the input document concepts to form a chain. In fact, concepts in the document can form a graph in any arbitrary shape. In addition, we attempt to build a generative language model contrary to the most dominant application of CRFs applied to discriminative problems. In other words, we are not interested in learning the best weights for diverse features that converge to the maximum value over a training dataset, instead, given the semantic relatedness between the observed concepts, we are interested in finding the probability that a query concept is generated from a specific document.

As an illustrative example, consider the query $q = \{\text{Journalism}\}$ and the document d that is composed of the following paragraph, which is selected from Document LA082290-0094 of TREC CD5:

> Singleton, [...], bought the Star-News for $55 million from the Knight-Ridder newspaper chain in June 1989.

Figure 2a shows the representation of the query and the document based on their concepts and semantic relatedness relationships. As seen in the figure, four concepts 'Knight Ridder,' 'William Dean Singleton,' 'Newspaper,' and 'Star-News' have been spotted in the document. Also, the concept 'Journalism' has been found in the query. Dashed lines show semantic relatedness between the query concept and document concepts, and solid lines represent semantic relatedness between document concepts. In this figure, concepts correspond to the Wikipedia articles with the same names and semantic relatedness are found using a semantic analysis system that estimates relatedness between Wikipedia entries.

This example highlights two main challenges of representing documents and queries based on their semantic concepts, which we address in the following. First, contrary to the bag-of-words model, where the probability of generating a query term given a document is estimated based on its occurrence in the document and in the collection, here we need to model semantic relatedness between query concepts and document concepts. We represent relatedness relations as probability dependencies. In our model, two semantically related concepts are modeled as dependent neighbors and two not semantically related concepts are modeled as non-neighboring nodes, which are independent given all other concepts. For forming this graph, our model relies on semantic analysis systems that measure semantic relatedness

between concepts in documents. These systems usually provide semantic relatedness score for pairs of concepts, where those with a score more than a specific threshold are considered to be semantically related.

Second, for a document of size n concepts, finding all semantic relatedness relationships is of order $O(n!)$. Given that such relatedness relations represent probability dependencies, finding the probability distribution over documents is quite complex and hardly possible for a big corpus. Our approach addresses this problem by avoiding finding the distribution over the input variables; hence, it is a good choice for estimating the probability of output variables (query concepts), without worrying about the joint distribution of input variables (document concepts). To be more clear, as we will see in Sect. 3.3, SELM uses conditional random fields for calculating the conditional probability of a query concept given document concepts. Based on this probability calculation approach, the relationships between document concepts have no impact on the conditional probability of generating query concepts and can be ignored in estimating the rankings of documents. We also assumed that semantic relationships between query concepts and document concepts are in the form of one-to-one correspondence. Figure 2b shows relationships in the example depicted in Fig. 2a that are used by SELM for the estimation of semantic rankings.

As seen in Fig. 2b, semantic similarities between concepts 'Knight Ridder' and 'Newspaper,' 'Newspaper' and 'William Dean Singleton,' 'William Dean Singleton ' and 'Star-News,' and 'Newspaper' and 'Star-News' have no impact on the rank of document 'd' given query 'q.' The details of ranking algorithms are presented in the next section.

3.3 Proposed model

We let $G = (V, E)$ be an undirected graph, where $V = D \bigcup Q$ and D be a set of document variables whose values are observed for any input document and Q be a set of query variables whose values need to be predicted by the model. Document and query variables correspond to concepts found in documents and queries, respectively. Document and query variables take binary values of (0,1), where the value of 1 indicates that the corresponding concept exists in a given document or query. The random variables are connected by undirected weighted edges, E, showing their degree of semantic relatedness. We denote an assignment to D by D_d, and an assignment to Q by Q_q. According to this model, a query concept Q_{q_j} is an assignment to Q in which the values of all variables except the jth variable are zero. The value of the jth element is 1. In this work, we assume that query concepts have no dependencies to each other. Hence, for a query $q = \{q_1, ..q_n\}$, $P(q|d) = \prod_{j=1}^{n} P(q_j|d)$. There are seminal works in the literature that consider dependencies between query terms in retrieval models [30]. Nonetheless, analyzing dependencies between query concepts is not the subject of this work and we leave it for future work.

As an example, the sample query and document that is depicted in Fig. 2 can be represented as an undirected graph $G(V, E)$, where $V = D \bigcup Q$, and D is the set of nodes corresponding to all entities of the knowledge base where nodes corresponding to the concepts 'Knight Ridder,' 'William Dean Singleton,' 'Newspaper,' and 'Star-News' get the value of 1, while all other nodes get value of 0. Also, Q consists of nodes corresponding to all entities in the knowledge base, while the value of all nodes except the node corresponding to 'Journalism,' that is equal to 1, is zero. In this example, E represents edges between 'Journalism' and the document nodes according to Fig. 2b.

In order to generate a ranking score for documents given a query term q_j, a scoring function needs to be defined based on the interpolation of two probabilities: the probability of the query given the document expressed as $P_{\text{selm}}(Q_{q_j}|D_d)$, and the probability of the

query given the collection of all documents denoted by $P(Q_{q_j}|\text{Col})$. The scoring function is formulated as:

$$\begin{aligned}\text{Score}_{\text{selm}}(d, q) &= P(Q_q|D_d) \\ &\simeq \sum_{j=1}^{|q|} \log P(Q_{q_j}|D_d)\end{aligned} \tag{1}$$

where according to the Jelinek–Mercer [56] interpolation function, we have:

$$P(Q_{q_j}|D_d) = \begin{cases} (1-\lambda)P_{\text{selm}}(Q_{q_j}|D_d) + \lambda P(Q_{q_j}|\text{Col}) & \text{similar concept found} \\ \lambda P(Q_{q_j}|\text{Col}) & \text{Otherwise} \end{cases} \tag{2}$$

Based on this model, we wish to find $P_{\text{selm}}(Q_{q_j}|D_d)$, the probability of a given query concept based on a given document. According to [23], we have:

$$P_{\text{selm}}(Q_{q_j}|D_d) = \frac{1}{Z(D_d)} \exp\left(\sum_{i=1}^{i=k} f_i(C_i, q_j, D_d)\right) \tag{3}$$

where $C_i \subseteq V$ is a clique over G and $C_i \not\subset D$, f_i is a feature function defined over C_i. $Z(d)$ is a normalization factor and is defined as:

$$Z(D_d) = \sum_j \exp\left(\sum_{i=1}^{i=k} f_i(C_i, Q_{q_j}, D_d)\right) \tag{4}$$

Q has $|Q|$ different assignments in each of which a node has a value of 1 and the others have the value of 0. The partition function Z is the sum of the non-normalized probability for all of $|Q|$ possible query concepts. Based on our definition of feature functions, which we will introduce in the following paragraph, the value of $f_i(C_i, Q_{q_j}, D_d)$ is zero for those concepts in Q that are not semantically related to concepts of d. Given d has n concepts and each of them are maximally related to m query concepts, Z can be computed by the summation of at most $n \times m$ non-normalized probabilities.

Based on the query term independence assumption, there is no edge between the $|Q|$ query nodes. Hence, a C_i has exactly one node from Q. Considering this fact, we may have three types of features: (1) features defined over document concepts, (2) featured defined over a set that includes one query concept and an arbitrary number of document concepts, and finally (3) features defined over a pair of a query concept and a document concept. The first set of features appears both in the non-normalized probability and Z in Eq. (3); therefore, they will cancel each other out in the normalized probability. Therefore, we do not need to consider them for estimating the score measure. In this paper, we also avoid calculating the second possible set of features because of its induced complexity and instead, we focus on the third set of features. It means that in our example in Fig. 2, we do not define a feature over the set {'Knight Ridder,' 'Newspaper,' 'Journalism }. Instead, we define two features {'Newspaper,' 'Journalism } and {'Knight Ridder,' 'Journalism }. Based on our assumptions, each C_i is a two-node clique that has one node from Q and one node from D that are connected through an edge, expressing that two corresponding concepts are semantically related to each other. Given $C_i = (x, y)$, $x \in D$, a node in the document space, $y \in Q$, a node in the query space, and the value of x and y is assigned by d and q_j, the feature function f_i is defined as follows:

$$f_i(C_i, q_j, D_d) = \begin{cases} \text{SemRel}(x, y) & x_d = y_{q_j} = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

where SemRel(x, y) is the value of semantic relatedness between two concepts associated with x and y. Now, the probability of $P(Q_{q_j}|\text{Col})$ is defined based on the document probabilities and collection statistics as follows:

$$P(Q_{q_j}|\text{Col}) = \frac{\sum_{d_i \in \text{Col}} P_{\text{selm}}(Q_{q_j}|D_d)}{|\text{Col}|} \tag{6}$$

Returning to our example, the score generated for the document depicted in Fig. 2 for the query 'Journalism' is equal to the probability of assigning the value of 1 to the Journalism node, given that the value of 1 is assigned to four nodes 'Knight Ridder,' 'William Dean Singleton,' 'Newspaper,' and 'Star-News.' This probability is defined using the features estimated over the following four sets: {'Knight Ridder,' 'Journalism'}, {'William Dean Singleton,' 'Journalism'}, { 'Newspaper,' 'Journalism' }, and {'Star-News,' 'Journalism'}. The values of features are defined based on semantic similarities found by any semantic analysis systems.

In Sect. 4, we will see how the results generated by SELM can be integrated with traditional keyword-based retrieval models for generating better results.

4 Integration module

Recalling Fig. 1, the results generated by SELM are fed into the integration module in order to be interpolated with the results obtained from keyword-based systems. As we will show later in the experimental results section, while SELM and other retrieval models can produce overlapping results, in many cases a subset of their relevant and correct results is distinct and non-overlapping. For this reason, the interpolation of these models can benefit from the correctly retrieved documents of each model and hence lead to improved performance.

Integrating different language models for finding a combined similarity score has been a topic of research in the recent years. In [3], a model is proposed to integrate language model θ_D, which is a language model based on the term dependency assumption, and $\theta_{\bar{D}}$, which is a language model based on non-dependency assumption in the following form:

$$\begin{aligned} P(q|d) &= \prod_{i=1}^{n} P(q_i|d) \\ &= \prod_{i=1}^{n} [P(q_i, \theta_D|d) + P(q_i, \theta_{\bar{D}}|d)] \\ &= \prod_{i=1}^{n} [P(q_i, |d)p(\theta_D|d) + P(q_i, |d)p(\theta_{\bar{D}}|d)] \\ &= \prod_{i=1}^{n} [P(q_i, |d)\lambda_{\theta_D} + P(q_i, |d)\lambda_{\theta_{\bar{D}}}] \end{aligned} \tag{7}$$

In this mode, λ_{θ_D} and $\lambda_{\theta_{\bar{D}}}$ are the probability of choosing dependency or non-dependency models given a document. The last line reformulates the model as a mixture model where λ_{θ_D} and $\lambda_{\theta_{\bar{D}}}$ are mixture weights and needed to be estimated. For integrating SELM and other language models, we follow a similar approach but with important differences:

$$\text{Score}(d, q) = \lambda_{KW} \tilde{\text{Score}}_{KW}(d, q) + \lambda_{\text{selm}} \tilde{\text{Score}}_{\text{selm}}(d, q) \tag{8}$$

where $\text{Score}_{\text{selm}}(d, q) = P_{\text{selm}}(Q_q|D_d)$, and $\tilde{\text{Score}}$ means a normalized score. Similar to Eq. (7), we use a linear mixture model that has mixture weights for combining different language models. On the other hand, we did not integrate probabilities in query term level, instead we integrate scores that are found over the whole query. The reason is that there is no shared interpretation of query terms across different models: in SELM, queries are interpreted as a set of concepts, each of which are associated with one or more query terms. Our integration model is close to what is proposed in [13] and [27], especially the *CombSUM* combination formula, according to which the scores of multiple systems are added together for creating the final score of a document. We also use the normalization strategy exploited in [27] for normalizing scores before integrating them. According to this strategy, the normalized score is defined as follows:

$$\tilde{\text{Score}}(d, q) = \frac{\text{Score}(d, q) - \text{MinScore}}{\text{MaxScore} - \text{MinScore}} \tag{9}$$

where MinScore and MaxScore are the minimum and maximum scores among the retrieved documents.

The integration model in Eq. 8 allows us to integrate semantic scoring with any arbitrary scoring. We use the EM algorithm to estimate mixture weights. For each query q, $\theta_q = \{\lambda_{\theta_{KW}}, \lambda_{\theta_{\text{selm}}}\}$, we have:

$$\theta_q^* = \arg\max_{\theta_q} \log\left(\sum_{i=1}^{i=N} \lambda_{\theta_{KW}} \text{Score}_{KW}(d, q) + \lambda_{\theta_{\text{selm}}} \text{Score}_{\text{selm}}(d, q)\right) \tag{10}$$

where N is the total number of documents and $\lambda_{\theta_{KW}} + \lambda_{\theta_{\text{selm}}} = 1$. In order to estimate λ, the mixture weight for a given query q is computed as follows:

$$\lambda_{\theta_{KW}}^t = \frac{1}{N} \sum_{i=1}^{i=N} \frac{\lambda_{\theta_{KW}}^{t-1} \text{Score}_{KW}(d_i, q)}{\lambda_{\theta_{KW}}^{t-1} \text{Score}_{KW}(d_i, q) + \lambda_{\theta_{\text{selm}}}^{t-1} \text{Score}_{\text{selm}}(d_i, q)} \tag{11}$$

The mixture weight is calculated for each query separately, making it possible to assign different weights to semantic- and keywords-based models for retrieving different queries. To terminate the EM iterations, we set a threshold such that changes less than the threshold will stop the EM algorithm. In our experiments, we find that EM converges quickly usually converging in less than 5 iterations.

Returning to the example depicted in Fig. 2, the final score that is calculated for the document is estimated based on the score found by a keyword-based retrieval system, which calculates ranking scores based on document and query term matching, and the SELM probability function that takes into account similarities between the query concept 'Journalism' and document concepts.

The following example from our experiments clarifies the impact of the interpolation of SELM with other keyword-based models. For the Trec topic 340: 'Land Mine Ban,' the state-of-the-art techniques such as [30] would not be able to retrieve documents that do not explicitly include the keywords such as *land, land mine*, or *ban* but are relevant to the query from a content perspective, e.g., FBIS3-44701 is ranked 398 by [30] because it does not have the explicit query keywords while it is a relevant document to the query in the gold standard. However, SELM retrieves this document and ranks it in the first position. The

interpolation of SELM + SDM proves to be effective in that this relevant document is ranked in position 9.

5 Query expansion for implementing SELM over semantic indices

Query expansion, expanding an original query with additional words for the purpose of expressing user intent more effectively, has been widely explored in the literature, and very successful results have been reported so far [54]. Recently, knowledge-enabled query expansion techniques, i.e., automatic methods that utilize knowledge expressed in sources like Wikipedia and Freebase for query expansion, have been introduced and implemented [7,52].

In this section, we investigate SELM more thoroughly and show that although it is defined as a language model for retrieval, it can be implemented by expanding queries not with words or texts but with a set of related concepts.

Given a query, for being matched and selected for ranking, a document must have a conditional probability more than zero. Recalling Eq. (3) in Sect. 3.3, it means that there should be at least one feature f_i that is defined over cliques in the query and the document whose value is greater than zero. Referring to Eq. (5), it means that there should be at least one pair of concepts x and y whose semantic relatedness is greater than zero. Also recall that the concept indexer module of the semantic retrieval framework (Fig. 1) indexes concepts found by the semantic annotation module for each document, but does not index the related concepts. Observably, it is difficult to search over concept indices for finding documents that do not have the exact query concepts but have the related ones.

As an example, consider the Trec query #324: 'Argentine/British Relations.' This query is annotated with a Wikipedia article with Wikipedia Id #16594665, which is named 'Argentina United Kingdom relations.' Also let d_1 be an arbitrary document that is annotated with just one concept: the Wikipedia article #82533, which is named 'International relations.' In the concept index produced by concept indexer module in Fig. 1, there is a posting list associated with concept #82533 that includes document d_1. Assuming that the concept 'International relations' is semantically related to 'Argentine/British Relations,' finding d_1 for this query over the concept index is a challenge that needs to be addressed.

For addressing this issue, we employ a simple yet effective approach that logically produces the same set of results but is much simpler to implement given the structure of inverted indices that we have used. According to this approach, we expand a given query with all concepts that are semantically related to its concepts, using semantic relatedness measures as a coefficient. We pose this new query against the semantic index and find matching documents. We rank the matching documents using SELM considering the original query. It should be noticed that although we expand the query for finding matching documents, we use the original query concepts in our ranking module. In our example, we pose query '16594665 OR (0.2)82533' against the semantic index meaning that those documents that contain concept #16594665 match the query and return value 1 for the feature defined over the clique (16594665, 16594665). But those documents that contain concept #82533 match the query and return value 0.2 for the feature defined over the clique (16594665 , 82533). In the following, we show why the results of the expanded query are the same as the original formation of SELM.

Without loss of generality and based on the query term independence assumption, let us assume c_k is the query concept found in q (the other concepts will be treated independently). Further assume, $C = \{c_i\}\ 0 \leq i \leq m$ are semantically related concepts to c_k found by the

semantic analysis module, where m is the total number of concepts in the knowledge base and i be any number between 0 and m. Let the degree of similarity be measured as $\{r_i\}$ $0 \leq i \leq m$ where r_i are real numbers between 0 and 1. The expanded query will be C, where each adding concept c_i has a effectiveness coefficient equal to its corresponding r_i. For any arbitrary document with concepts $D = \{c_j\}\, 0 \leq i \leq m$, we can define feature functions over binary cliques (c_i, c_j), when $c_i \in C$ and $c_j \in D$. Based on our approach, we do not find all cliques between all possible pairs over C and D. Instead, we consider cliques over exactly matching pairs of concepts, i.e., all pairs in the form of (c_i, c_i) and the value for the feature defined over this pair is equal to its related r_i. The value of any feature function defined over (c_i, c_i) in this expanded model is equal to the value of the feature function defined over (c_k, c_i) in the original model. Hence, the probability estimated by Eq. (3) is exactly the same in both cases. Clearly our expansion method is a way around implementing SELM over concept indices and is identical with the original language model presented in Sect. 3.

6 Different configurations of our framework

In this section, we describe three configurations of the proposed semantic retrieval framework. These configurations differ in their semantic analysis module, which measures the degree of similarity between concepts. Figure 3 shows the three different configurations of the framework. As seen in this figure, Wikipedia is the knowledge base that is used as the underlying source for entities. We chose to use the Tagme entity linking engine to play the role of the semantic annotation module in these configurations. The choice of this annotation engine was motivated by a recent study reported in [6] that showed that Tagme was the best

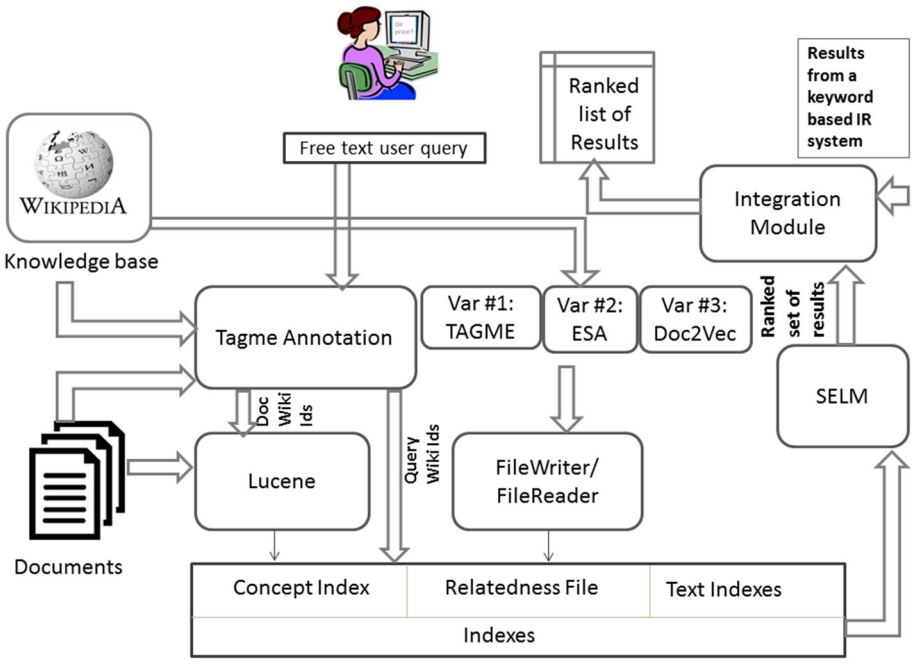


Fig. 3 Three configurations of the semantic retrieval framework

performing annotation system on a variety of document types such as Web pages and Tweets, and also has publicly accessible RESTful API and is available as an open source project.

For indexing concepts identified in each document, we use their corresponding ConceptIDs, which is an integer number corresponding to the ID of a Wikipedia entry, as a key in Lucene.[3] The concept indexer module has a second stage in which the normalization factor Z (Eq. 4) is calculated and stored for each document. The normalization factor is calculated based on the degree of semantic relatedness between concepts of a document and all of the concepts of the collection. In the semantic analysis module, which provides the required semantic relatedness values, we use three different techniques that are the basis for the three different configurations. All three semantic analysis techniques save their relatedness estimations in files that are loaded by concept indexer and SELM for indexing, ranking and retrieval.

6.1 Tagme semantic analysis

Tagme [12], which is known for its on-the-fly entity linking service, also provides an entity relatedness measuring service. Inspired by the work presented in [50], Tagme calculates relatedness between two Wikipedia pages using their shared links to other Wikipedia pages and produces a number between 0 and 1. We directly used this service and generated similarities between all concept pairs in our concept index.

6.2 ESA semantic analysis

Explicit semantic analysis (ESA) [9] is a well-known method for finding semantic similarities between natural language texts. ESA represents the meaning of a text by mapping it to a weighted vector of Wikipedia entries, known as '*concept vector*,' and exploits cosine similarity for finding similarities between vectors. Since ESA is designed to find similarities between texts but not knowledge base entries, it cannot be directly used in our framework. For this purpose, we represent a Wikipedia page by the text part of the 'dbo:abstract' and 'rdfs:comment' fields of its corresponding DBPedia entity, so for each pair of concepts in concept index, it is possible to find their corresponding concept vectors and calculate their similarities.

6.3 Paragraph2Vec semantic analysis

Representing words in a vector space using neural networks has emerged recently as one of the successful semantic modeling techniques for texts [31,47]. Word vectors are learnt to represent semantics of words, i.e., semantically close words such as 'powerful' and 'strong' are mapped to close points in the multi-dimensional space where the representation of semantically unrelated words such as 'powerful' and 'pears' is more distant [26]. Based on the vector representation of words, paragraph-to-vector is proposed in [26] to map the meanings of variable-length texts to vectors. Our third configuration of the framework uses paragraph vectors to represent Wikipedia entries and find the degree of their relatedness. For forming paragraph vectors, we use word vectors that were trained over Wikipedia. Paragraph vectors are trained over concept texts. Similarly to the second configuration, we used the text part of the 'dbo:abstract' and 'rdfs:comment' fields of the corresponding DBPedia entities as the text of each Wikipedia concept.

[3] http://lucene.apache.org/.

Table 1 TREC collections used in our experiments

Collection	Documents	Topics
Robust04	528,155	301–450, 601–700
ClueWeb09-B	50,220,423	1–200
ClueWeb12-B	52,343,021	1–50

7 Experiments

In this section, we describe experiments for analyzing the performance of the proposed semantic retrieval framework.

7.1 Experimental setup

In our experiments, we adopted three widely used document collections: (1) TREC Robust04, (2) ClueWeb09-B (TREC Category B, which is the first 50 million English pages of the ClueWeb09 corpora), and (3) ClueWeb12-B (the TREC 2013 Category B subset of the ClueWeb12 corpora). Table 1 summarizes the datasets and the queries that were used in our experiments. As explained in Sect. 6, we chose to annotate document collections using the Tagme entity linking engine. As a part of its results, Tagme provides a confidence value for each retrieved concept. We use Tagme's recommended confidence value of 0.1 for pruning unreliable annotations. As suggested in [7] and due to limited computational resources, we do not entity link all documents in the ClueWeb09-B and ClueWeb12-B document collections. Instead, we pool the top one hundred documents from all of the baseline text retrieval runs. The top 100 documents retrieved from all of our baselines along with their annotations as well as their runs and their evaluation metric results are made publicly accessible.[4]

In these experiments, we use Jelinek-Mercer [56], which is the linear interpolation of the document language model and the collection language model with coefficient λ set to 0.1.

The queries that were used in the experiments are the title fields of 250 Trec topics for Robust04, 200 Trec Web track topics for ClueWeb09-B, and 50 Web track topics for ClueWeb12-B. In our model, both queries and documents are required to be modeled as a set of concepts. For ClueWeb09-B queries, we use the Google FACC1 data that provide explicit annotations for the Web track queries. These annotations include descriptions and sub-topics from which we use the description annotations. For Robust04 and ClueWeb12-B queries, there are no publicly available annotations. For our experiments, we employ Tagme with a confidence value of 0.25. We found a number of missing entities and also annotation errors in the results. As an example, Topic 654, 'same-sex schools,' was annotated as 'Homosexuality,' and 'Catholic School,' which are inconsistent. We manually revised these annotations to fix several errors. In this case, our revised annotation was the concept 'Single-sex education' for the topic number 654. All query annotations made by Tagme and also revisions are publicly available in the earlier mentioned Git repo.

7.2 Baselines

For the sake of comparison, we chose the sequential dependence model (SDM) [30], which is a state-of-the-art retrieval model based on Markov random field that assumes dependencies between query terms. In addition, we compare SELM with two query expansion models: a

[4] https://github.com/SemanticLM/SELM.

variant of relevance model (RM3) [25], and entity query feature expansion (EQFE) [7]. RM3 extracts the relevant terms and uses them in a combination with the original query. RM3 is known to improve the retrieval performance over methods that do not use expansion terms. EQFE is an expansion method that enriches the query with features extracted from entities found in queries, entity links to knowledge bases, and the entity context. It has already been shown [7] that EQFE improves retrieval performance significantly over the state-of-the-art methods. In this paper, and to keep our experiments comparable to these methods, we used the parameter settings reported in [7,25] for the baseline methods. SELM is interpolated with these three baseline systems based on Eq. (8) in order to form three variations, referred to as SELM + SDM, SELM + RM3, and SELM + EQFE.

7.3 Results

In this section, we report the performance of SELM and its interpolation with baseline methods. For the purpose of this evaluation, we conducted two series of experiments: First, we thoroughly evaluate SELM, when it is configured to use Tagme as its semantic analysis module. In these experiments, we aim at evaluating the effect of semantic retrieval in improving the performance of the other baseline retrieval models.

In SELM, each query concept has a similarity threshold $0 < \alpha < 1$, such that all similarities less than α are pruned, i.e., concepts with similarities less than α are considered as unrelated to the query concepts. In this set of experiments, α is determined using 10-fold cross-validation and is optimized for mean average precision (MAP) effectiveness.

Second, we conduct a set of experiments to compare the performance of SELM under three different configurations, where it is configured to use Tagme, ESA, and Para2Vec similarity measurement techniques (See Sect. 6). The purpose of these experiments is to compare the impact of different semantic similarity measurement techniques on our proposed semantic retrieval framework.

For each collection and in all experiments, we report the mean average precision (MAP), precision at rank 20 (P@20), and normalized discounted cumulative gain at rank 20 (nDCG@20). The statistical significance of differences in the performance of SELM models with respect to other retrieval methods is determined using a paired t test with a confidence level of 5%. For evaluating ClueWeb09-B and ClueWeb12-B, the relevance judgments of the whole corpus have been used.

7.3.1 Performance evaluation

SELM Interpolation Effectiveness Table 2 presents the evaluation results on three datasets. The interpolation of SELM with all baselines improves their performance. SELM + SDM outperforms SDM significantly across two measures: MAP and nDCG@20 on all datasets (up to + 9.2% MAP and + 6.1% nDCG@20). Also, SELM + SDM improves P@20 compared to SDM over Robust04, ClueWeb09-B and outperforms SDM significantly over ClueWeb12-B (up to + 5.7% P@20). SELM + RM3 outperforms RM3 across all measures on all datasets (up to + 5.5% MAP, + 6.1% nDCG@20, and + 7.9% P@20). The improvements are statistically significant on P@20 over ClueWeb12-B, MAP over Robust04 and ClueWeb12-B, and on nDCG@20 on all datasets. SELM + EQFE outperforms EQFE on all metrics for all datasets, and the observed improvements are statistically significant for ClueWeb09-B.

Success/failure analysis Figure 4 provides analysis of queries whose effectiveness are improved/hurt by the variants of the SELM method. For the sake of clarity and easier visu-

Table 2 Evaluation results for the interpolation of SELM with the three baseline methods

	Map	Δ	*p* value	P@20	Δ	*p* value	nDCG @20	Δ	*p* value
Robust04									
SDM	0.2615			0.3715			0.4235		
SELM + SDM	0.2858†	+9.2	0.0001	0.3811	+2.5	0.1419	0.4405†	+4	0.0136
RM3	0.2937			0.388			0.434		
SELM + RM3	0.31†	+5.5	0.0003	0.3986	+2.6	0.0577	0.4501†	+3.6	0.0061
EQFE	0.3278			0.3797			0.4237		
SELM + EQFE	0.3382†	+3	0.0197	0.3902	+2.7	0.1465	0.4353	+2.7	0.1233
ClueWeb09-B									
SDM	0.1143			0.3412			0.21467		
SELM + SDM	0.1183†	+3.4	0.0156	0.3495	+2.4	0.7	0.22793†	+6.1	0.006
RM3	0.12			0.3447			0.22108		
SELM + RM3	0.123	+2.5	0.0699	0.3477	+0.8	0.6	0.23411†	+5.9	0.006
EQFE	0.1096			0.3184			0.2119		
SELM + EQFE	0.117†	+6.7	0.0004	0.3298†	+3.5	0.0475	0.23078†	+8.9	0.0004
ClueWeb12-B									
SDM	0.0421			0.209			0.12679		
SELM + SDM	0.0446†	+5.1	0.002	0.221†	+5.7	0.0019	0.13407†	+5.6	0.0025
RM3	0.0359			0.189			0.11098		
SELM + RM3	0.038†	+5.5	0.0122	0.204†	+7.9	0.0001	0.11776†	+6.1	0.0042
EQFE	0.0469			0.232			0.14633		
SELM + EQFE	0.0493	+4.8	0.0535	0.234	+0.8	0.5	0.14981	+2.3	0.2

Statistical significance is shown by †
Relative difference percentage and *p* values from paired *t* test are shown as Δ% and *p* value

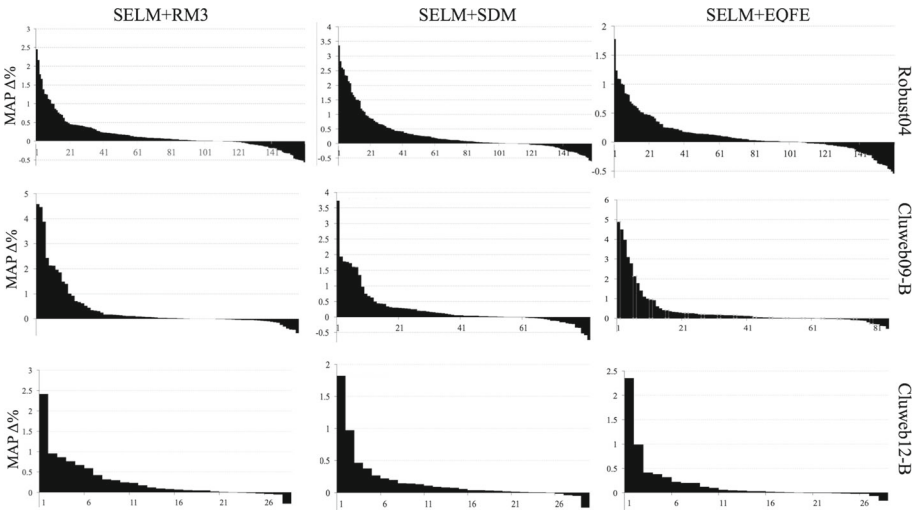


Fig. 4 MAP Δ% of interpolated SELM & baselines (e.g., SELM + SDM vs. SDM). Positives show improvement over baseline

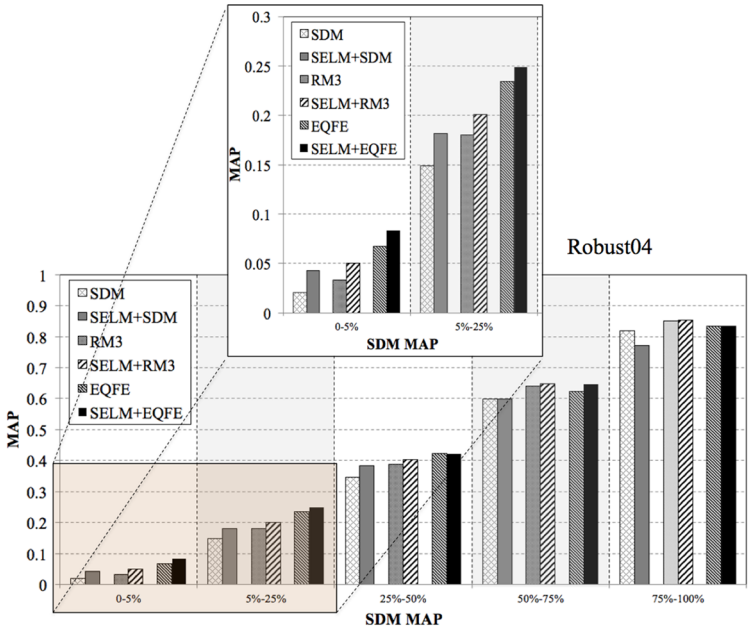
Table 3 The number of queries helped by SELM variants

	R'04	CW'09	CW'12
SELM + SDM versus SDM	115	59	25
SELM + RM3 versus RM3	113	50	23
SELM + EQFE versus EQFE	107	62	19

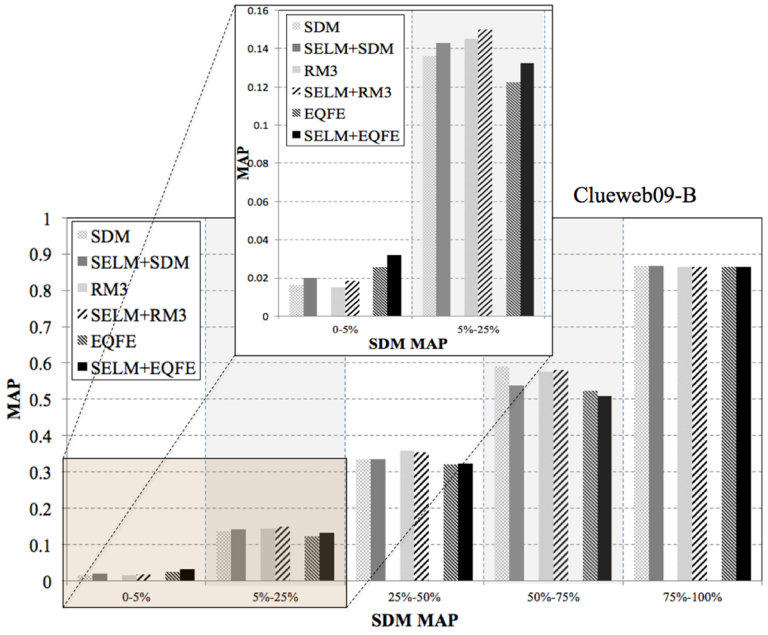
alization of the results in this figure, we did not include top queries that resulted in dramatic improvements. For instance, for the SELM + SDM model on the Robust dataset, we did not include the top two queries that were helped almost 90 times and 6 times, respectively, compared to SDM. In these figures, the relative percentage improvement of MAP for SELM + SDM over SDM, SELM + RM3 over RM3, and SELM + EQFE over EQFE is reported. Given the fact that SELM returns no results for queries with no concepts, we only consider the queries that have at least one concept annotation, which is equal to 163 queries for Robust04, 94 and 34 for ClueWeb09-B and ClueWeb12-B, respectively. As outlined in Table 3, out of the 163 queries for the Robust04 dataset, SELM + SDM helps 115, SELM + RM3 helps 113, and SELM + EQFE helps 107 of the queries. In ClueWeb09-B and for the 94 queries, SELM+SDM helps 59, SELM + RM3 helps 50, and SELM + EQFE helps 62 queries. For ClueWeb12-B and the associated 34 queries, SELM + SDM helps 25, SELM + RM3 helps 23, and SELM + EQFE helps 19 queries. All the help/hurts were determined by comparing the relative difference percentage of MAP of an interpolated SELM method compared to its respective baseline. SELM + SDM is the method that has seen a high improvement in terms of the number of helped queries. The reason can be due to the fact that SDM, contrary to RM3 and EFQE, is a method that has not been augmented by expansions from documents or knowledge base data and links. Hence, it can benefit the most when combined with the semantic perspective that is offered by SELM.

We also analyze SELM variants with regard to their effect on a range of easy to difficult queries. For this analysis, we divide queries into buckets of MAP ranges according to their MAP from the SDM baseline. Queries that have larger SDM MAPs are considered to be easier queries compared to the ones that have a lower SDM MAP, which are those that we will consider to be more difficult. Figure 5 illustrates this analysis. The figure has three parts for each of the document collections. In the figure, a SELM variant is paired with its associated baseline, e.g., SELM + SDM and SDM, to show how much improvement was obtained as a result of the interpolation. In addition, we have provided a zoomed-in view of the results for the most difficult queries in order to be able to clearly depict the improvement made on such queries. This analysis shows that SELM is effective in improving the more difficult queries. For Robust04, all queries except the easiest queries (queries whose SDM MAP are between 75 and 100%) are improved by all SELM interpolated methods compared to their respective baselines. In ClueWeb09-B, all difficult queries (MAP$<$ 50%) have been improved and specially more difficult queries (MAP$<$ 25% as shown in the zoom) have received noticeable improvement. SELM performed well on the ClueWeb12-B collection, where all of the queries, specially the difficult queries, were improved.

Table 4 shows that the interpolation of SELM with baselines outperforms the baselines across all measures for the most difficult queries. We considered queries whose MAP value for the SDM baseline is less than 0.05 to be the *most difficult* queries. As an instance, query #92 ('the wall') is a difficult query for the keyword-based system (SDM MAP $=$ 0.0009). Keyword-based query expansion cannot help much (RM3 MAP $=$ 0.0009). Even EQFE (EQFE MAP $=$ 0.0008), which uses semantic knowledge for query expansion is



(a)



(b)

Fig. 5 Mean retrieval effectiveness across different query-difficulties measured according to the percentile of SDM

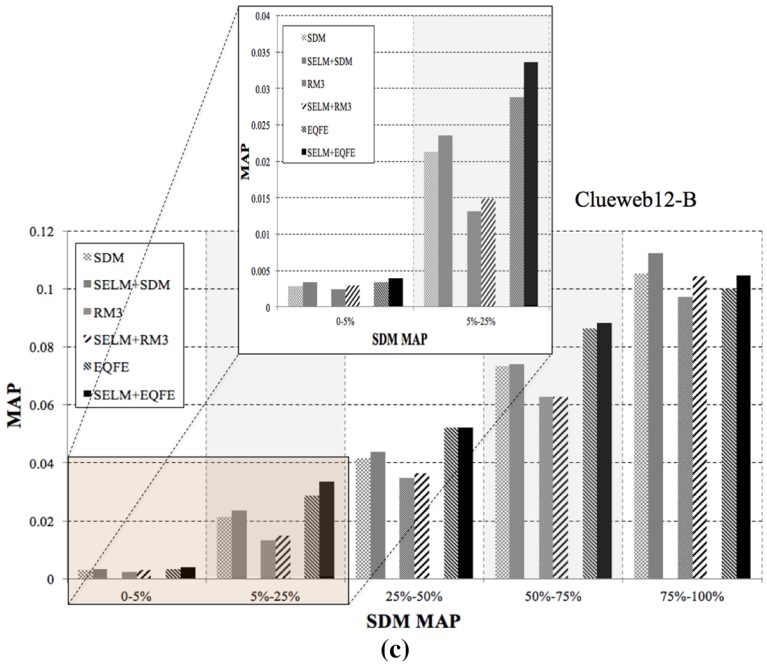


Fig. 5 continued

far behind SELM (SELM + SDM MAP = 0.0234, SELM + RM3 MAP = 0.0231, and SELM + EQFE MAP = 0.14). SELM works with query annotations (The Pink Floyd album named 'the wall') for retrieval, which helps SELM to search within documents that have concepts related to music, rock bands, and Pink Floyd. Hence, SELM has a better chance of finding related documents. On the other hand, SELM faces difficulties when dealing with search queries that are annotated with general concepts. As an example, none of the SELM interpolations produce effective results for the query #44('map of the united states'). This is because the query is annotated with one concept only, i.e., United States, which is a very general concept with relationships to a lot of unrelated entities irrelevant to the topic of the query. As another example, web query #142 ('Illinois state tax') produces poor results when processed by SELM variants. This query is annotated with only one concept (Illinois), which is a general concept with a lot of diverse relationships, and at the same time does not cover the main topic of the query, which is taxes. We hypothesize that more effective query annotation techniques that are able to find both *relevant* and *specific* concepts that relate to the core topic of the query will help improve SELM. There is a progressive body of work in the literature that focus on query analysis and segmentation [16,34]. We leave verification of this hypothesis and application of the query analysis literature to our future work.

Analysis of interpolation success The main premise of our work was that the semantic-enabled model would retrieve documents that would not be otherwise retrieved by the other baseline models. This has been empirically shown in Fig. 6. The three sub-figures show the comparative analysis of the retrieval of distinct relevant documents retrieved by SELM compared to the other methods. As seen, for all three datasets, SELM retrieves a significant number of relevant documents that are missed by the other methods (shown in the Venn

Table 4 Comparison of the retrieval models on the most difficult queries (SDM MAP< 5%)

	Map	Δ	*p* value	P@20	Δ	*p* value	nDCG	Δ	*p* value
Robust04	Difficult queries (Map: 0–5%), number of queries: 42								
SDM	0.0196	111%†	0.008	0.0817	20%	0.089	0.0896	34%†	0.080
SELM + SDM	0.0415			0.0987			0.1203		
RM3	0.0309	55%†	0.030	0.0707	45%†	0.029	0.0778	44%†	0.048
SELM + RM3	0.0480			0.1024			0.1117		
EQFE	0.0637	24%	0.07	0.0951	21%†	0.033	0.0983	29%	0.1
SELM + EQFE	0.0794			0.1158			0.1272		
ClueWeb-09	Difficult queries (Map: 0–5%), number of queries: 85								
SDM	0.0164	22%†	0.039	0.0779	9%	0.1	0.0512	18%†	0.035
SELM + SDM	0.0201			0.0845			0.0607		
RM3	0.0154	20%	0.05	0.0690	15%†	0.021	0.0445	22%†	0.025
SELM + RM3	0.0186			0.0797			0.0546		
EQFE	0.0258	28%†	0.030	0.1125	7%	0.07	0.0844	18%†	0.007
SELM + EQFE	0.0323			0.1202			0.0999		
ClueWeb-12	Difficult queries (Map: 0–5%), number of queries: 34								
SDM	0.0185	6.7%	0.1	0.0893	20%†	0.017	0.0533	6%	0.1
SELM + SDM	0.0197			0.100			0.0565		
RM3	0.0131	7%†	0.022	0.0727	11%†	0.02	0.0371	8%†	0.04
SELM + RM3	0.0141			0.0803			0.0402		
EQFE	0.0237	11%	0.2	0.1257	6%	0.09	0.0736	3.2%	0.4
SELM + EQFE	0.0257			0.1333			0.0761		

†shows statistical significance using a paired t test ($\alpha < 0.05$)

diagrams). The bar charts show the number of distinct non-overlapping relevant documents retrieved by SELM that have not been observed in any of the other approaches within the top-10 results (the x-axis shows queries and is ordered in descending order.). This shows how SELM is effective in the retrieval process and why its integration improves the overall performance.

7.3.2 Evaluation of semantic retrieval framework

In this section, we investigate how different semantic analysis modules affect the performance of the semantic retrieval framework. Figure 7 shows the performance of three configurations of the semantic retrieval framework on Robust04, measured by their MAP, where SELM uses Tagme (SELM V1), ESA (SELM V2), and Para2Vec (SELM V3) for finding similarities between concepts, respectively. In order to use ESA, we did not calculate similarities between all pairs of concepts, due to heavy processing needed and limited resources. Instead, we find 'concept vectors' (See Sect. 6.2 for concept vectors) for each concept, and find similarities between that concept and the top 500 concepts in its concept vector. We assumed that the concept has no relatedness to the other concepts that are not in its top 500 concepts list. For Para2Vec, we used the Gensim library and its Doc2Vec model [43]. For training the model, we set the window size to 8, and the dimensionality of the feature vectors to 50.

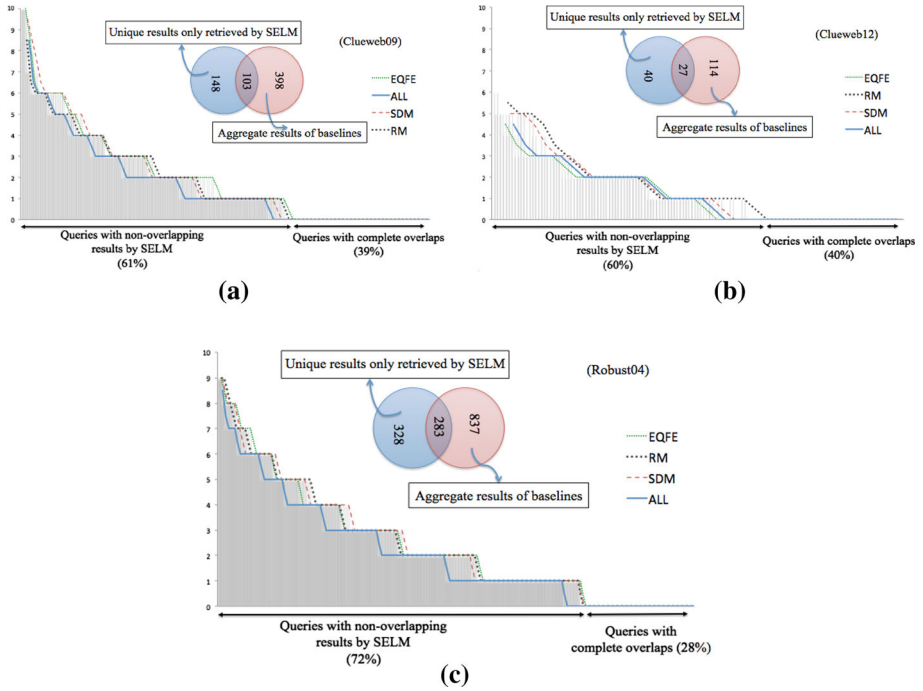


Fig. 6 Comparison of the distinct results of SELM compared to the other methods

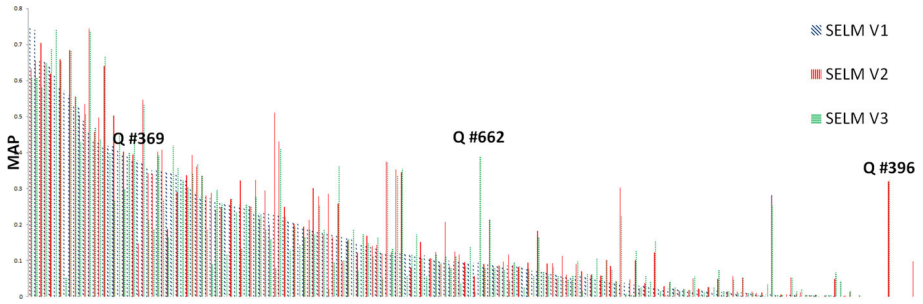


Fig. 7 SELM variations MAP on the Robust dataset, queries sorted by their MAP values with SELM V1

In Fig. 7, queries are sorted based on their MAP values for SELM V1, and for all similarity modules, all concepts with similarity less than 0.85 are pruned as unrelated.

As seen in this figure, despite similarities, there are distinguishing queries whose MAP are completely different in different configurations. For example in this figure, queries #396, and #662 are among queries with good performance with SELM V2 and SELM V3, while they have poor performance with SELM V1. On the other hand, MAP values of SELM V1 for query #369 are considerably higher than theirs with SELM V2 and V3. For having a better comprehension of semantic retrieval configuration differences, let us look at these queries more closely. Query #396 is annotated with one concept, which is the Wikipedia Entry named 'Sick building syndrome.' Tagme finds 'Evaluation (workplace), 0.9721,' 'Employee monitoring, 0.9721,' 'Induction training, 0.971,' and 'Induction programme, 0.971' as top

similar concepts to this one, where all of them enjoy similarities of more than 0.97. For the same query, ESA finds 'Building code, 0.4962,' 'Field Building (Chicago), 0.4177,' 'Public Service Building (Portland, Oregon), 0.4058,' and 'Olympia Centre, 0.375' as the top related concepts, all of them with a relatedness score of less than 0.42. Also, Para2Vec provides 'Sound stage, 0.6366,' 'Molwyn Joseph, 0.6189,' 'Halas Hall, 0.6059,' as its top concepts related to the query concept. In this example, all similar concepts provided by ESA and Para2Vec are pruned by SELM because they are less than the threshold of 0.85. On the other hand, Tagme, which provides a proportionally better list, shows a poor performance because of the high values it assigns to its related concepts.

The same pattern repeats for Query #662, with the concept 'Telemarketing,' that is found to be similar to 'Broadcast law, 0.8896,' 'Media scrum, 0.8894,' 'Media regulation, 0.8891,' 'Transfer (propaganda), 0.8891,' an others by Tagme, while ESA finds 'NTT DoCoMo, 0.3876,' 'Cold calling, 0.3235,' 'Assisted GPS, 0.3134,' 'Flip (form), 0.3129' as the list of related concepts and Para2Vec finds 'Secure messaging, 0.6839,' 'Peterborough railway station,0.66,' 'PeaZip, 0.6053' as its top list of related concepts. Although Tagme is doing a better job in finding similar concepts, it gives them high values that decrease the performance of SELM. However, Tagme similarity analysis is not always negatively impacting the performance. For example, query #369, which includes concept 'Anorexia nervosa,' is found to be similar with 'Bulimia nervosa, 0.9115,' 'Eating disorder, 0.8848,' 'Eating disorder not otherwise specified, 0.8568,' 'Intermittent explosive disorder, 0.8495' and others by Tagme. This query performs considerably better with SELM V1 than SELM V2 (ESA) which found 'Anorexia mirabilis, 0.3737' as related and SELM V3 (Para2Vec) that found 'Valence (psychology), 0.9221,' 'False pregnancy, 0.9177,' 'Emotional security, 0.9173,' as the top related concepts.

This example highlights importance of the similarity thresholds in the performance of the various configurations. Table 5 shows the performance of SELM where it uses Tagme (V1), ESA (V2), and Para2Vec (V3) for measuring similarities between concepts over Robust04, ClueWeb09-B, and ClueWeb12-B datasets. In this experiment, we set similarity threshold, (α), to three different values, 0.3, 0.5, 0.85, where all concepts with similarities less than this threshold are pruned as unrelated. In this table, each measure is calculated as the average value of those queries that have at least one annotated concept.

As seen in this table, ESA is the best performing system across all measures with different α values over Robust and ClueWeb09-B datasets, and its performance on MAP is statistically significant compared to the other configurations. On ClueWeb12-B, no system has a significant lead on MAP under different thresholds, but Para2Vec has a better P@20 compared to the other two for $\alpha = 0.3$. ESA has a strict similarity measurement that assigns low values to less related concepts. For example, for Robust dataset, where $\alpha = 0.85$, ESA finds 1.35 related concepts for any concept on average. This is much less than Tagme (with 20.76 average similar concepts per concept) and Para2Vec (with 29.59 average similar concepts per concept). For Robust and ClueWeb09-B, this strict similarity measurement leads to a better performance and for ClueWeb12-B it is not significantly harmful across the MAP measure. The other observable fact from this table is that Para2Vec has an almost identical performance in two settings where $\alpha = 0.5$ and $\alpha = 0.3$. The reason is that a big portion of the similarities generated by Para2Vec in our experiments are larger than 0.5. It makes filtering values less than 0.3 and 0.5 to produce almost identical lists.

We also analyze these configurations with regard to their performance on difficult queries. For this analysis, we used their MAP from the SDM baseline. Queries that have SDM MAPs less than 0.05 are considered as the difficult bucket. Figure 8 shows this analysis. The figure has three parts for each dataset. In each part of the figure, MAP values for these three

Table 5 Evaluation results for SELM variations with three thresholds (α)

SELM	Map	p value	P@20	p value	nDCG@20	p-val
Robust04						
	$\alpha = 0.85$					
V1	0.1555		0.2243		0.2558	
V2	$\mathbf{0.1867}^{\dagger}_{v1,v3}$	0.0001_{v1} 0.0001_{v3}	$\mathbf{0.2710}^{\dagger}_{v1,v3}$	0.0001_{v1} 0.0071_{v3}	$\mathbf{0.3018}^{\dagger}_{v1,v3}$	0.0001_{v1} 0.0347_{v3}
V3	0.1678		0.2532		0.2532	
	$\alpha = 0.5$					
V1	0.1100		0.2010		0.2230	
V2	$\mathbf{0.1772}^{\dagger}_{v1,v3}$	0.0001_{v1} 0.0001_{v3}	$\mathbf{0.2603}^{\dagger}_{v1,v3}$	0.0001_{v1} 0.0001_{v3}	$\mathbf{0.2901}^{\dagger}_{v1,v3}$	0.0001_{v1} 0.0001_{v3}
V3	0.1045		0.1953		0.2373	
	$\alpha = 0.3$					
V1	0.1150		0.2166		0.2389	
V2	$\mathbf{0.1646}^{\dagger}_{v1,v3}$	0.0001_{v1} 0.0001_{v3}	$\mathbf{0.2489}^{\dagger}_{v1,v3}$	0.0144_{v1} 0.0001_{v3}	$\mathbf{0.2781}^{\dagger}_{v1,v3}$	0.005_{v1} 0.0001_{v3}
V3	0.1042		0.1948		0.2363	
ClueWeb09-B						
	$\alpha = 0.85$					
V1	0.0644		0.2131		0.1342	
V2	$\mathbf{0.0715}^{\dagger}_{v3}$	0.7_{v1} 0.37_{v3}	0.2196		0.1345	0.98_{v1} 0.86_{v3}
V3	0.0628		0.207		0.1325	
	$\alpha = 0.5$					
V1	0.0561		0.1853		0.1237	
V2	$\mathbf{0.0745}^{\dagger}_{v1,v3}$	0.019_{v1} 0.0054_{v3}	0.2232	0.0607_{v1} 0.0538_{v3}	0.1348	0.4148_{v1} 0.5176_{v3}
V3	0.0577		0.1929		0.127	

Table 5 continued

SELM	Map	*p* value	P@20	*p* value	nDCG@20	p-val
	$\alpha = 0.3$					
V1	0.0581		0.2035		0.1254	
V2	$\mathbf{0.0751}^{\dagger}_{v1,v3}$	0.019_{v1} 0.0037_{v3}	0.2232	0.2577_{v1} 0.0546_{v3}	0.1344	0.447_{v1} 0.5444_{v3}
V3	0.0577		0.1929		0.1271	
ClueWeb12-B						
	$\alpha = 0.85$					
V1	0.03		0.1428		0.0834	
V2	0.0309	0.6029_{v1} 0.6057_{v3}	0.1471		0.0810	
V3	0.03		0.15	0.7236_{v1} 0.7678_{v2}	0.0896	0.39_{v1} 0.1239_{v2}
	$\alpha = 0.5$					
V1	0.0229		0.1214		0.0663	
V2	0.02810		0.1428		0.0737	
V3	0.0302	0.067_{v1} 0.459_{v2}	$\mathbf{0.1628}^{\dagger}_{v1}$	0.0280_{v1} 0.0848_{v2}	0.0900	0.1310_{v1} 0.0582_{v2}
	$\alpha = 0.3$					
V1	0.0232		0.1242		0.0629	
V2	0.0278		0.14		0.0717	
V3	0.0302	0.1001_{v1} 0.4526_{v2}	$\mathbf{0.1628}^{\dagger}_{v1,v2}$	0.0312_{v1} 0.0403_{v2}	$\mathbf{0.09}_{v2}$	0.0517_{v1} 0.0228_{v2}

Statistical significance and p-values to a variation v shown by $\dagger_v$ and $p\text{value}_v$

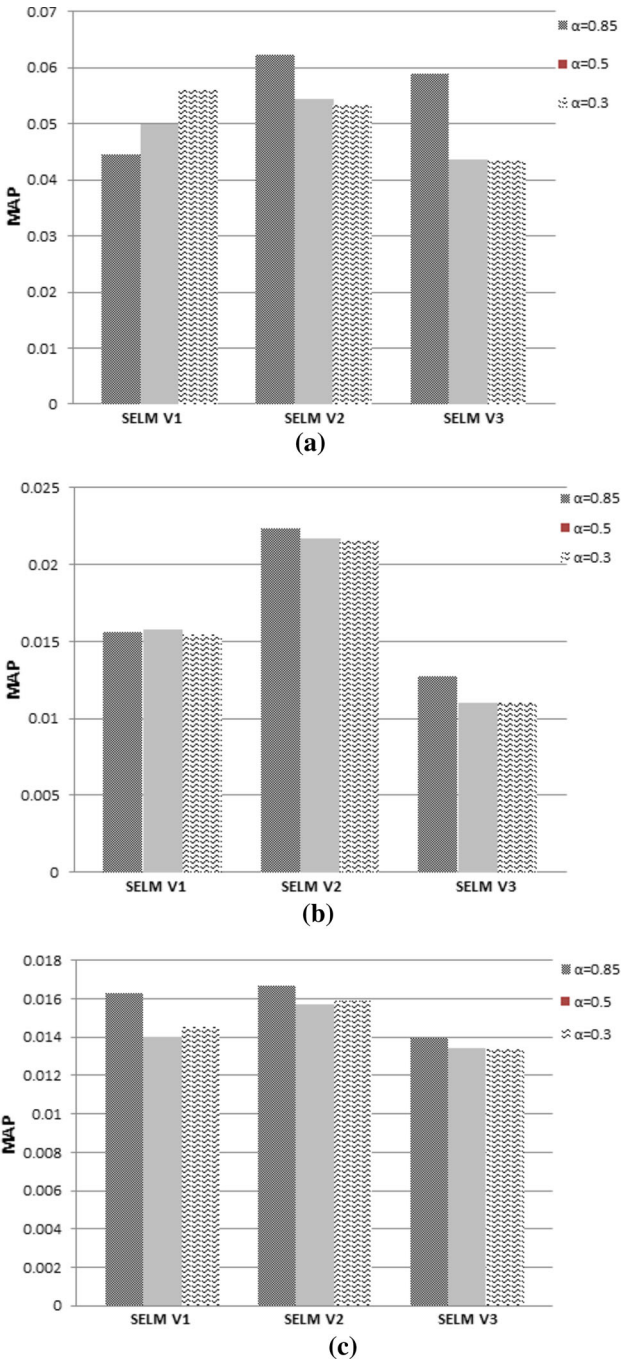


Fig. 8 SELM variations MAP values for difficult queries, over Robust, ClueWeb09-B, and ClueWeb12-B collections under three similarity thresholds: $\alpha = 0.85$, $\alpha = 0.5$, $\alpha = 0.3$

configurations are illustrated. MAP values are calculated as the average MAP over the difficult set of queries. This analysis shows that SELM V2 (ESA) is the most effective system in answering the difficult queries over all datasets. For Robust04, SELM V2 ($\alpha = 0.85$) is the best performing system (MAP = 0.062 compared to V1 MAP, which is 0.056, and V3 MAP, which is 0.058), while V2 in all thresholds has larger MAP than its peers V1 and V3. The same patterns repeats for ClueWeb09-B, where SELM V2 has the largest MAP values (MAP = 0.022 for $\alpha = 0.85$) compared to the best values obtained by other configurations (V1 MAP = 0.015, V2 MAP = 0.0127). In this dataset, SELM V2 in all of its thresholds is better than the other systems in all of their similarity thresholds. SELM V2 is the best performing configuration over ClueWeb12-B as well. When $\alpha = 0.85$, SELM V2 is slightly better than SELM V1 and considerably better than SELM V3. For all other similarity thresholds, SELM V2 is the leading configuration in this collection.

For the last experiment, we analyze how different similarity thresholds affect the performance of SELM variations. For this purpose we run SELM with 13 different similarity thresholds, ranging from 0.3 to 0.95 with intervals of 0.05. Figure 9 illustrates this experiment. This figure has three parts, each of which shows SELM variations performance measured by their MAP with different similarity thresholds (shown as α) over a document collection. For this experiment, MAP is calculated over all queries including those that have no concepts attached. These queries have no answer over all SELM variations for all α values. As seen in Fig. 9a, SELM V2, the configuration that uses ESA for semantic similarity, has the best MAP over all thresholds (α values) in Robust04 collection. SELM V2 enjoys a slight improvement as α increases, while SELM V1 and V3 experience sharp improvements with higher values for α. In ClueWeb09-B (Fig. 9b), SELM V2 keeps its advantage over V1 and V3, though the MAP chart has a different pattern. In this dataset, SELM V2 has a slight decrease in its MAP values as α increases, while SELM V1 and SELM V3 MAPs fluctuate over α, with a tendency to mostly increase after $\alpha = 0.55$. In ClueWeb12-B (Fig. 9c), three variations have a very similar performance after $\alpha = 0.8$, while SELM V3 has a slight lead to the others prior to that point. From Fig. 9, we can observe SELM V2 keeps a steady performance over different thresholds. However, the best working threshold differs for each document collection and each method, and hence can be found by learning methods with a set of training data.

8 Related work

Semantic modeling and retrieval have gained the attention of diverse research communities in recent years. Latent semantic models and statistical translation language models are two examples of ranking models that propose alternative ways for representing texts other than the classic bag-of-words representation for capturing semantics of documents and queries. In latent semantic models such as [2,17], documents and queries are modeled as a set of words generated from a mixture of latent topics, where a latent topic is a probability distribution over the terms or a cluster of weighted terms. Through these models, the similarity between a query and a document is analyzed based on their corresponding latent topics. In translation language models, the likelihood of translating a document to a query is estimated and used for the purpose of ranking [19,22]. In these models, translation relationships between a term in the document and a term in the query are estimated, and because a term in the document can be translated into a different term in the query, these models can be utilized to cope with the vocabulary gap problem. Contrary to latent semantic models and translation models, in SELM, documents and queries are not modeled using latent semantics. This fact introduces

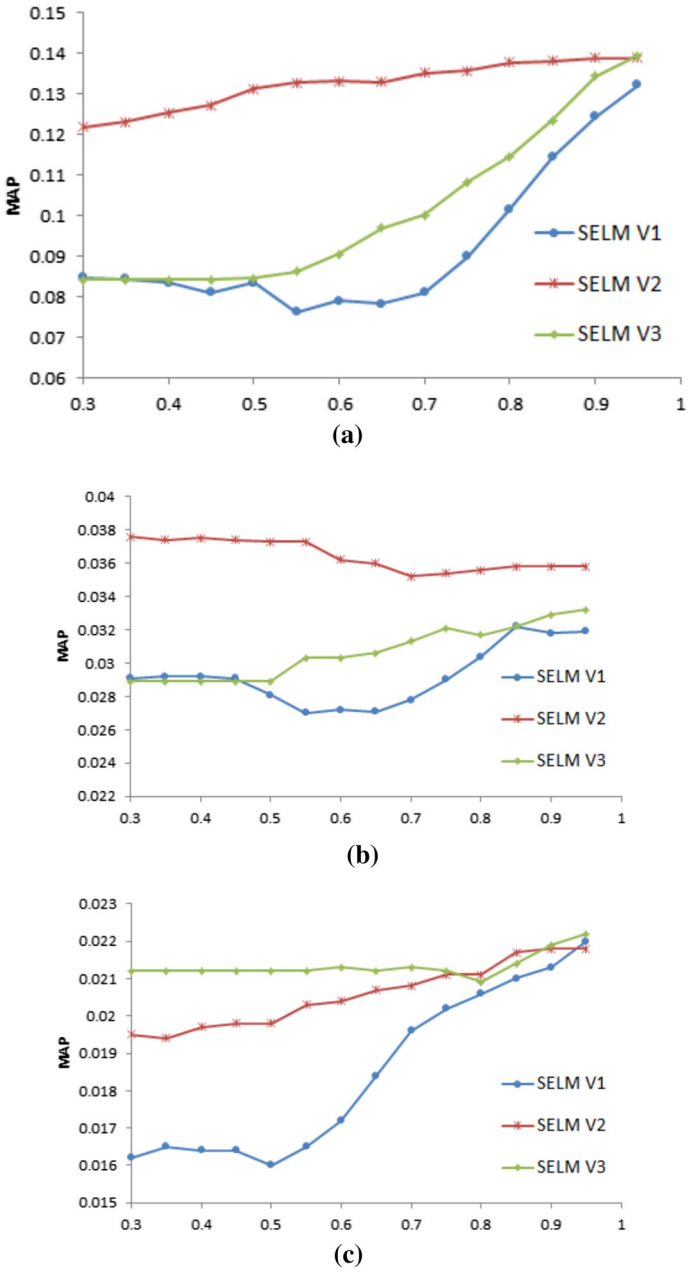


Fig. 9 SELM variations MAP values for difficult queries, over Robust, ClueWeb09-B, and ClueWeb12-B collections under three similarity thresholds: $\alpha = 0.85$, $\alpha = 0.5$, $\alpha = 0.3$

more flexibility for SELM as it is not dealing with estimating topics for documents or learning pairwise relationships between query and document words, instead, it exploits concepts and their degree of semantic relatedness from state-of-the-art semantic linking systems.

Exploiting general or domain-specific knowledge in retrieval has been extensively studied in the literature. Vallet et al. [48] propose using knowledge that is formally represented in domain ontologies for enhancing domain-specific search. In their approach, a free text query is translated to RDQL, a query language for RDF, and posed over a formally represented domain ontology. A related document is one that is annotated with instances of the result tuples. The amount and quality of information that is modeled within the ontology limits the performance of ontology-based retrieval systems. On the other hand, Wikipedia and Freebase are two comprehensive sources of general world knowledge that are used as alternatives to domain-specific ontologies. In [45], the authors present methods for indexing documents with Wikipedia concepts and representing documents with bag of concepts. These concepts are interlingual, hence can be used for cross-lingual retrieval. Similarly, the work in [9] provides a bag-of-concept representation for documents based on the notion of concept vectors from explicit semantic analysis (ESA). This work embeds a set of feature selection methods into its retrieval process in order to handle the noisy nature of the concept representation. Both [45] and [9] use ESA representation of concepts for the purpose of concept ranking and retrieval. Contrary to [9] and [45], our work is not attached to a specific knowledge representation framework and can work with any semantic annotation and analysis system. In [42] and [53], documents and entities are presented as a bag-of-entities, contrary to the classic bag-of-words representation. Similar to the approach presented in our paper, entities are produced by entity linking systems. In these works, documents are ranked based on the number of times that query entities are observed in the documents, and contrary to our approach semantically related entities are not considered for finding relevant documents. The semantic retrieval framework presented in our paper is basically motivated differently, where it is designed to be used in situations where there is no exact match between queries and documents and ranking and scoring can only be done based on a shared semantic space. In [52] and [7], Freebase and Wikipedia are used for expanding query terms. In these methods, object descriptions and category classifications are used among other information resources for enriching queries. We used [7] as one of our baselines and compared its performance with variants of SELM.

Learning-to-rank methods, which construct ranking models for documents based on training data [28], are another direction of work in semantic retrieval where semantic knowledge is incorporated into training and building models. In [20], a learning-to-rank approach is provided for predicting and ranking related news. According to this approach, documents are annotated with references to named entities (consisting of organizations, persons, and locations), and named entity features are used for learning a ranking model. Two samples of entity features that are used are 'string similarity between entity names' and 'the number of sentences in a document in which an entity is a subject.' The learning-to-rank method described in [44] uses convolutional neural networks for learning the embedding of queries and short documents (assumed to be sentences) into low-dimensional vector space, and then uses query and document vectors for learning a ranker model. It is discussed in [44] that the low-dimensional vector space representation preserves semantic aspects of queries and documents, and hence can be employed for semantic matching. EsdRank [51] is a learning-to-rank technique that provides a basis for using semi-structured meta-data in ranking models. It models entities from semi-structured external data as objects that connect queries and documents. Query-object and object-document features are defined to link queries to these entities and to rank documents based on entities. Samples of query-object features are annotator confidence, and BM25 scores between query and entity text fields and an instance of document-object feature is the BM25 scores between document and entity text fields. Contrary to learning-to-rank approaches, our semantic retrieval system is a generative language model that does not need training data for learning a model.

There is a body of work in the literature that study semantic search in terms of searching over semantic data which is represented as semi-structured or structured documents. Swoogle [8] is a well-known search engine for searching over RDF and OWL documents. Swoogle finds RDF and OWL documents, indexes them, and answers queries. Given the fact that RDF repositories may contain up to billions of RDF triples, the work in [55] addresses the efficiency challenge in searching over semantic data. In [46], a method is proposed to find a suitable combination of Linked, RDF, and structured data resources for an input query. In [15], a hybrid search system is proposed that enables search over both text and semi-structured ontologies such as RDF triples. This system provides a clean and simple user interface where users can pose queries over semi-structured data without a need to know SPARQL [41] or other RDF-based query languages. These works contribute to search over semantic data, which differs form our contribution, i.e., using semantic data for improving ad hoc retrieval.

Another related topic of research to our work is retrieving entities from documents. In [21], Wikipedia is used as a pivot for searching, and Wikipedia categories and their relations are used as the main source for entity retrieval. Zhiltsov et al. [57] propose to generalize the sequential dependence model for structured documents such as DBPedia. In their model, a mixture of language models is employed for retrieving entities that are represented in a five-field scheme, which is designed for DBpedia entities. In [35], user logs are analyzed for finding implicit user feedback in the context of entity search. The other impressive works in this area include but are not limited to [1,5]. In [58], a language modeling approach is proposed to integrate multiple document features such as PageRank, indegree, and URL length for entity search. The main focus of all these works, which is returning an entity or a list of entities for user queries, is different from the research goal of our work, which is the utilization of knowledge represented in knowledge bases, such as Wikipedia, for document retrieval.

9 Conclusion

In this paper, we have proposed a semantic retrieval framework for ad hoc queries. This framework includes a semantic-enabled language model as its main component, which represents documents and queries through a graph of concepts, where the relatedness of a query to a given document is calculated based on the semantic relatedness of their concepts. We have provided three different configurations of the framework, where semantic relatedness between query and documents concepts are calculated based on three different strategies. We conducted comprehensive experiments for evaluating the performance of the proposed framework under these configurations, and compared its performance under different parameter settings. We also analyzed the impact of the interpolation of the semantic language model with different keyword-based systems. Our empirical evaluations show that our proposed model can complement and enhance the performance of keyword-based systems and its interpolation with other retrieval models can significantly improve their performance from the perspective of various IR measures.

For future work, we seek to enhance the semantic retrieval framework by modeling dependencies between query entities, by exploring other semantic analysis systems, and by investigating the impact of different query types on semantic retrieval. Currently, the retrieval model presented in this work assumes that query entities are independent. Nonetheless, semantically related entities to a query entity can be affected by other query entities. For

example, in the query 'Obama Family Tree,' the semantically related entities to 'Obama' are different when it is coming with 'Family' comparing to other cases, e.g., when it is coming with entities related to presidential campaign. Modeling dependencies between query entities can improve retrieval performance in queries that have more than one entity.

In this paper, we thoroughly analyzed three different semantic analysis systems and their impact on the performance of our retrieval framework. We would like to complement our work with exploring other semantic relatedness systems, especially new methods in finding similarities between entities using embedding techniques [18,49].

As we showed in the evaluation section, different spotting and linking methods can affect the performance of semantic retrieval system. In future work, we would like to utilize different query analysis techniques for anticipating the performance of semantic retrieval on various queries. That is especially helpful in the interpolation process in order to appropriately weight keyword-based or semantic-based scores.

Acknowledgements This work is partially funded by Ferdowsi University of Mashhad Grant Number 2/39715.

References

1. Billerbeck B, Demartini G, Firan C, Iofciu T, Krestel R (2010) Exploiting click-through data for entity retrieval. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 803–804
2. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J. Mach. Learn. Res. 3:993–1022
3. Cao G, Nie J-Y, Bai J (2005) Integrating word relationships into language models. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 298–305
4. Chakrabarti S, Kasturi S, Balakrishnan B, Ramakrishnan G, Saraf R (2012) Compressed data structures for annotated web search. In: Proceedings of the 21st international conference on World Wide Web. ACM, pp 121–130
5. Cheng T, Yan X, Chang KC-C (2007) Entityrank: searching entities directly and holistically. In: Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment, pp 387–398
6. Cornolti M, Ferragina P, Ciaramita M (2013) A framework for benchmarking entity-annotation systems. In: Proceedings of the 22nd international conference on World Wide Web. ACM, pp 249–260
7. Dalton J, Dietz L, Allan J (2014) Entity query feature expansion using knowledge base links. In: Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval. ACM, pp 365–374
8. Ding L, Finin T, Joshi A, Pan R, Cost RS, Peng Y, Reddivari P, Doshi V, Sachs J (2004) Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM, pp 652–659
9. Egozi O, Markovitch S, Gabrilovich E (2011) Concept-based information retrieval using explicit semantic analysis. ACM Trans Inf Syst (TOIS) 29(2):8
10. Ensan F, Bagheri E (2017) Document retrieval model through semantic linking. In: Proceedings of the tenth ACM international conference on web search and data mining, WSDM 2017, Cambridge, United Kingdom, February 6–10, 2017, pp 181–190
11. Ferragina P, Scaiella U (2010) 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, pp 1625–1628
12. Ferragina P, Scaiella U (2012) Fast and accurate annotation of short texts with wikipedia pages. IEEE Softw 29(1):70–75
13. Fox EA, Shaw JA (1994) Combination of multiple searches. NIST Special Publication SP, pp 243–243
14. Gabrilovich E, Markovitch S (2007) Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th international joint conference on artificial intelligence, IJCAI'07. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, pp 1606–1611
15. Gärtner M, Rauber A, Berger H (2014) Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation. Knowl Inf Syst 41(3):761–792

16. Guo J, Xu G, Cheng X, Li H (2009) Named entity recognition in query. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. ACM, pp 267–274
17. Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pages 50–57. ACM
18. Ji G, He S, Xu L, Liu K, Zhao J (2015) Knowledge graph embedding via dynamic mapping matrix. In: ACL (1), pp 687–696
19. Jin R, Hauptmann AG, Zhai CX (2002) Language model for information retrieval. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 42–48
20. Kanhabua N, Blanco R, Matthews M (2011) Ranking related news predictions. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 755–764
21. Kaptein R, Serdyukov P, De Vries A, Kamps J (2010) Entity ranking using wikipedia as a pivot. In: Proceedings of the 19th ACM international conference on information and knowledge management. ACM, pp 69–78
22. Karimzadehgan M, Zhai C (2010) Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In: Proceedings of the 33rd ACM SIGIR, pp 323–330
23. Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth ICML, pp 282–289
24. Lashkari F, Ensan F, Bagheri E, Ghorbani AA (2017) Efficient indexing for semantic search. Expert Syst Appl 73:92–114
25. Lavrenko V, Croft WB (2001) Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 120–127
26. Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In: ICML, vol 14, pp 1188–1196
27. Lee JH (1997) Analyses of multiple evidence combination. In: ACM SIGIR forum, vol 31, pp 267–276
28. Liu T-Y et al (2009) Learning to rank for information retrieval. Found Trends Inf Retr 3(3):225–331
29. McCallum A, Bellare K, Pereira F (2012) A conditional random field for discriminatively-trained finite-state string edit distance. arXiv:1207.1406
30. Metzler D, Croft WB (2005) A markov random field model for term dependencies. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 472–479
31. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. *arXiv preprint* arXiv:1301.3781
32. Miller DR, Leek T, Schwartz RM (1999) A hidden markov model information retrieval system. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 214–221
33. Milne D, Witten IH (2013) An open-source toolkit for mining wikipedia. Artif Intell 194:222–239
34. Mishra N, Saha Roy R, Ganguly N, Laxman S, Choudhury M (2011) Unsupervised query segmentation using only query logs. In: Proceedings of the 20th international conference companion on World wide web. ACM, pp 91–92
35. Mottin D, Palpanas T, Velegrakis Y (2013) Entity ranking using click-log information. Intell Data Anal 17(5):837–856
36. Ni Y, Xu QK, Cao F, Mass Y, Sheinwald D, Zhu HJ, Cao SS (2016) Semantic documents relatedness using concept graph representation. In: Proceedings of the ninth ACM international conference on Web search and data mining. ACM, pp 635–644
37. Otegi A, Arregi X, Ansa O, Agirre E (2015) Using knowledge-based relatedness for information retrieval. Knowl Inf Syst 44(3):689–718
38. Peng F, McCallum A (2006) Information extraction from research papers using conditional random fields. Inf Proces Manag 42(4):963–979
39. Pinto D, McCallum A, Wei X, Croft WB (2003) Table extraction using conditional random fields. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 235–242
40. Ponte JM, Croft WB (1998) A language modeling approach to information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 275–281
41. Prud E, Seaborne A, et al. (2006) Sparql query language for rdf

42. Raviv H, Kurland O, Carmel D (2016) Document retrieval using entity-based language models. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 65–74
43. Řehůřek R, Sojka P (2010) Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 workshop on new challenges for nlp frameworks. Valletta, Malta. ELRA, pp 45–50. http://is.muni.cz/publication/884893/en
44. Severyn A, Moschitti A (2015) Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 373–382
45. Sorg P, Cimiano P (2012) Exploiting wikipedia for cross-lingual and multilingual information retrieval. Data Knowl Eng 74:26–45
46. Tran T, Zhang L (2014) Keyword query routing. IEEE Trans Knowl Data Eng 26(2):363–375
47. Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, pp 384–394
48. Vallet D,.Fernández M, Castells P (2005) An ontology-based information retrieval model. In: The semantic Web: research and applications. Springer, pp 455–470
49. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph and text jointly embedding. In: EMNLP, vol 14, pp 1591–1601
50. Witten I, Milne D (2008) An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: Proceeding of AAAI workshop on wikipedia and artificial intelligence: an evolving synergy, AAAI Press, Chicago, USA, pp 25–30
51. Xiong C, Callan J (2015) Esdrank: Connecting query and documents through external semi-structured data. In: Proceedings of the 24th ACM international on conference on information and knowledge management. ACM, pp 951–960
52. Xiong C, Callan J (2015) Query expansion with freebase. In: Proceedings of the 2015 international conference on the theory of information retrieval, pp 111–120
53. Xiong C, Callan J, Liu T.-Y (2016) Bag-of-entities representation for ranking. In: Proceedings of the 2016 ACM on international conference on the theory of information retrieval. ACM, pp 181–184
54. Xu J, Croft WB (1996) Query expansion using local and global document analysis. In: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 4–11
55. Yuan P, Xie C, Jin H, Liu L, Yang G, Shi X (2014) Dynamic and fast processing of queries on large-scale rdf data. Knowl Inf Syst 41(2):311–334
56. Zhai C, Lafferty J (2001) A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 334–342
57. Zhiltsov N, Kotov A, Nikolaev F (2015) Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 253–262
58. Zhu J, Huang X, Song D, Rüger S (2010) Integrating multiple document features in language models for expert finding. Knowl Inf Syst 23(1):29–54

Faezeh Ensan is an Assistant Professor in the Department of Computer Engineering at Ferdowsi University of Mashhad, Iran. She is also an Honorary Research Associate at the University of New Brunswick, Canada. She received her Ph.D. in Computer Science from the University of New Brunswick, Canada in 2011 on Semantic Web technologies. Since then, she is working on applying Semantic Technologies in Information Retrieval. Her research interests include Semantic Web, Information Retrieval, Knowledge-based Systems and Semantic Search.

Weichang Du has been a Professor in Computer Science at University of New Brunswick, Canada since 1991. He obtained his M.Sc and Ph.D. in Computer Science from University of Victoria, Canada in 1985 and 1991. In past 25 years, he has published many research articles and supervised more than 50 Masters and PhD students. In recent years, he has been conducting research on designing and developing knowledge-based and intelligent software and knowledge systems and applications on Web and mobile platforms, including health-related systems and applications. He can be reached at wdu@unb.ca.