World Wide Web (2018) 21:1041–1067
DOI 10.1007/s11280-017-0468-7

Leveraging semantic resources in diversified query expansion

Adit Krishnan[1] · Deepak P.[2] · Sayan Ranu[3] · Sameep Mehta[4]

Received: 13 February 2017 / Revised: 25 April 2017 / Accepted: 9 May 2017 /
Published online: 5 June 2017

© The Author(s) 2017. This article is an open access publication

Abstract A search query, being a very concise grounding of user intent, could potentially have many possible interpretations. Search engines hedge their bets by diversifying top results to cover multiple such possibilities so that the user is likely to be satisfied, whatever be her intended interpretation. Diversified Query Expansion is the problem of diversifying query expansion suggestions, so that the user can specialize the query to better suit her intent, even before perusing search results. In this paper, we consider the usage of semantic resources and tools to arrive at improved methods for diversified query expansion. In particular, we develop two methods, those that leverage Wikipedia and pre-learnt distributional word embeddings respectively. Both the approaches operate on a common three-phase framework; that of first taking a set of informative terms from the search results of the initial query, then building a graph, following by using a diversity-conscious node ranking to

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering*
Guest Editors: Wojciech Cellary, Hua Wang, and Yanchun Zhang

✉ Deepak P.
deepaksp@acm.org

Adit Krishnan
aditk2@illinois.edu

Sayan Ranu
sayanranu@cse.iitd.ac.in

Sameep Mehta
sameepmehta@in.ibm.com

[1] Siebel Center for Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

[2] Queen's University Belfast, Northern Ireland, UK

[3] Department of Computer Science and Engineering, IIT Delhi, Hauz Khas New Delhi, 110016, India

[4] IBM-Research, New Delhi, 110070, India

prioritize candidate terms for diversified query expansion. Our methods differ in the second phase, with the first method Select-Link-Rank (SLR) linking terms with Wikipedia entities to accomplish graph construction; on the other hand, our second method, Select-Embed-Rank (SER), constructs the graph using similarities between distributional word embeddings. Through an empirical analysis and user study, we show that SLR ourperforms state-of-the-art diversified query expansion methods, thus establishing that Wikipedia is an effective resource to aid diversified query expansion. Our empirical analysis also illustrates that SER outperforms the baselines convincingly, asserting that it is the best available method for those cases where SLR is not applicable; these include narrow-focus search systems where a relevant knowledge base is unavailable. Our SLR method is also seen to outperform a state-of-the-art method in the task of diversified entity ranking.

Keywords Query expansion · Diversification · Semantic search · Wikipedia · Entity ranking

1 Introduction

Users of a search system may choose the same initial search query for varying information needs. This is most evident in the case of *ambiguous queries* that are estimated to make up one-sixth of all queries [30]. Consider the example of a user searching with the query *python*. It may be observed that this is a perfectly reasonable starting query for a zoologist interested in learning about the species of large non-venomous reptiles,[1] or for a comedy-enthusiast interested in learning about the British comedy group *Monty Python*.[2] However, search results would most likely be dominated by pages relating the programming language,[3] that being the dominant interpretation (aka *aspect*) in the Web. *Search Result Diversification (SRD)* [5, 37] refers to the task of selecting and/or re-ranking search results so that many *aspects* of the query are covered in the top results; this would ensure that the zoologist and comedy-fan in our example are not disappointed with the results. If the British group is to be covered among the top results in a re-ranking based SRD approach for our example, the approach should consider documents that are as deep in the un-diversified ranked list as the rank of the first result that relates to the group. In our exploration, we could not find a result relating to *Monty Python* among the first five pages of search results for *python* on Bing. Such difficulties in covering long tail aspects, as noted in [2], led to research interest in a slightly different task attacking the same larger goal, that of Diversified Query Expansion (DQE). Note that techniques to ensure coverage of diverse aspects among the top results are relevant for apparently unambiguous queries too, though the need is more pronounced in inherently ambiguous ones. For an unambiguous query: *python programming*, there are many aspects based on whether the user is interested in *books*, *software* or *courses*. Similarly, for another seemingly unambiguous query, *india*, the aspects of interest could include *railways*, *maps*, *news* and *cricket*.

DQE is the task of identifying a (small) set of terms (i.e., words) to extend the search query with, wherein the extended search query could be used in the search system to retrieve results covering a diverse set of aspects. For our *python* example, desirable top DQE

[1] https://en.wikipedia.org/wiki/Pythonidae

[2] https://en.wikipedia.org/wiki/Monty_Python

[3] https://en.wikipedia.org/wiki/Python_(programming_language)

expansion terms would include those relating to the programming language aspect such as *language* and *programming* as well as those relating to the reptile-aspect such as *pythonidae* and *reptile*. In existing work, the extension terms have been identified from sources such as corpus documents [34], query logs [21], external ontologies [2, 3] or the results of the initial query [34]. The aspect-affinity of each term is modeled either explicitly [21, 34] or implicitly [2] followed by selection of a subset of candidate words using the *Maximum Marginal Relevance (MMR)* principle [5]. This ensures that terms related to many aspects find a place in the extended set. Diversified Entity Recommendations (DER) is the analogous problem where the output of interest is a ranked list of entities from a knowledge base such that diverse query aspects are covered among the top entities.

In this paper, we consider the diversified query expansion problem and develop a three phase framework to exploit semantic resources for the problem. We use the framework to develop methods focusing on Wikipedia and pre-learned word embeddings respectively, leading to techniques that we call Select-Link-Rank (SLR) and Select-Embed-Rank (SER). Further, we outline how SLR can address diversified entity ranking, and illustrate that SER results can also be mapped to a corresponding DER result set.

Extension from WISE 2016 Paper In our WISE 2016 paper [18], we had proposed the SLR method. In this paper, we generalize SLR into a framework, and also develop another method based on the framework, SER, one targeted at exploiting pre-learned word embeddings. While this generalization and the new method remains the main extension to the earlier paper, we have added a significant number of empirical evaluations as well.

Our main contributions are:

- A three-phase skeletal framework targeted at exploiting semantic resources for diversified query expansion. This framework does not rely on query logs or other kinds of supervision, and thus, is immune to cold start issues.
- A Wikipedia-based grounding of the framework leading to a method, Select-Link-Rank, abbreviated *SLR*. SLR addresses both diversified query expansion and entity recommendation by harvesting terms from initial query results, followed by prioritizing terms and entities using the Wikipedia graph in a diversity conscious fashion.
- Select-Embed-Rank, abbreviated *SER*, another method based on the framework, but one that exploits word embeddings instead of Wikipedia. SER, like SLR, starts by selecting terms from initial query results, but constructs the graph using similarities of word embedding vectors, followed by a diversity ranking.
- We present an empirical evaluation including a user study that benchmark SLR and SER against the state-of-the-art methods for DQE and DER, illustrating the effectiveness of these methods over existing methods.

We survey related work in Section 2. This is followed by a concrete outline of the problem statement and solution framework in Sections 3.1 and 3.2 respectively. Sections 4 and 5 detail our DQE methods, SLR and SER, respectively. This followed by our empirical evaluation in Section 6 and conclusions in Section 7.

2 Related work

We will start by scanning the space of Search Result Diversification methods, followed by a detailed analysis of techniques for DQE/DER. This is followed by a brief overview of word embeddings, a semantic resource that one of our methods utilizes.

2.1 Search result diversification

Search Result Diversification is the task of producing a ranked result set of documents in a retrieval task such that most aspects of the query are covered. The pioneering SRD work [5] proposed the usage of the MMR principle in a technique that targets to reduce the redundancy among the top-results as a method to implicitly improve aspect representation:

$$\arg\max_{d} \ \lambda \times S_1(d, Q) - (1 - \lambda) \times \max_{d' \in S} S_2(d, d')$$

In MMR, the next document d to be added to the result set (S), is determined as that maximizing a score modeled as the relevance to the query (S_1) penalized by the similarity (S_2) to already chosen results in S. A more recent SRD method uses Markov Chains to reduce redundancy [37]. Since then, there have been methods to explicitly model query aspects and diversify search results using query reformulations [26], query logs [12] and click logs [16], many of which use MMR-style diversification.

2.2 Diversified query expansion

Diversified Query Expansion, a more recent task as well as the problem addressed in this paper, starts from a query and identifies a set of terms that could be used to extend the query that would then yield a more aspect-diverse result set; thus, DQE is the diversity-conscious variant of the well-studied Query Expansion problem [8]. In a way, DQE differs from SRD in being an *active* (or user-reliant) aspect diversification task targeted at providing some suggestions to the user so she can explicitly reformulate the query as needed; thus, this relaxes the SRD expectation that the system is capable of doing the diversification itself using just the initial query. Table 1 summarizes the various DQE methods in literature. Drawing

Table 1 Techniques for diversified query expansion

Method[a]	User Data Reqd.	External Resource Reqd.	Source of Exp. Terms	Remarks
BHN [2] (DER Baseline)	–	ConceptNet	Entity Names	Expansion terms from the small vocabulary of entity names
ts_{xQuAD} [34] (DQE Baseline)	Sub-topics (i.e, aspects) and sub-topic level relevance judgements	–	**Documents**	Relevance judgements are often impractical to get, in real systems
LBSN [21]	Query Logs	–	Query Logs	Cold start issue, also inapplicable for small-scale systems
BLN [3]	Query Logs	ConceptNet Wikipedia	Entity Names, Categories, Query Logs etc.	Expansion terms from small vocabulary as BHN and query log usage as LBSN
SLR (Ours)	–	Wikipedia	Documents	
SER (Ours)	–	Pre-learned Word Embeddings	Documents	

[a] When the authors have not used a name for a method, we will refer to it using the combination of first characters of author names.

inspiration from recent interest in linking text with knowledge-base entities (notably, since explicit semantic analysis [14]), BHN [2] proposes to choose expansion terms from the names of entities in the ConceptNet ontology, thus generating expansion terms that are focused on entities. BLN [3] extends BHN to use Wikipedia and query logs in addition to ConceptNet; the Wikipedia part relies on being able to associate the query with one or more Wikipedia pages, and uses entity names and representative terms as candidate expansion terms from Wikipedia. While such choices of expansion terms make BHN and BLN methods suitable for entity recommendations (i.e., DER), the limited vocabulary of expansion terms makes it a rather weak query expansion method. For example, though *courses* might be a reasonable expansion term for *python* under the computing aspect, BHN/BLN will be unable to choose such words since *python courses* is not an encyclopaedic concept to be an entity in the ConceptNet or Wikipedia. The authors in [3] note that the BLN-Wiki is competitive with BHN in cases where the query corresponds to a known Wikipedia concept, and that BHN performs better in general cases. We will use BHN as an entity ranking (DER) baseline in our experiments.

LBSN [21] gets candidate expansion terms from query logs. Such direct reuse of search history is not feasible in cold start scenarios and cases where the search engine is specialized enough to not have a large enough user base (e.g., single-user desktop search) to accumulate enough redundancy in query logs; our framework targets more general scenarios where query logs may not be available. ts_{xQuAD} [34], another DQE method, is designed to use terms from corpus documents to expand the query, making it immune to the small vocabulary problem and useful in a wide range of scenarios, much like the focus of SLR. However, ts_{xQuAD} works only for queries where the set of relevant documents are available at the aspect level. Given that, if each result document retrieved for the initial query may be deemed relevant to at least one aspect, a topic learner such as LDA [1] may be used to partition the results into topical groups by assigning each document to the topic with which it has the highest affinity. Since such topical groups are likely to be aspect-pure, such result partitions can be fed to ts_{xQuAD} to generate expansion terms without usage of relevance judgments. We will use the LDA-based ts_{xQuAD} as the baseline DQE technique for our experiments. Another related work is that of enhancing queries using entity features and links to entities [9], which may then be processed using search engines that have capabilities to leverage such information; we, however, target the DQE/DER problem where the result is a simple ordered list of expansion terms or entities.

2.3 Semantic resources for query expansion

We now consider research on using external semantic resources for query expansion. Due to the usage of Wikipedia and word embeddings in our method, we give a short summary of such resources and work on using such resources for query expansion.

2.3.1 Wikipedia

Wikipedia[4] is a free online encyclopaedia that allows collaborative editing of encyclopaedic articles. It contains an article associated with each entity it covers, and covers around five million entities overall. As already mentioned, BLN [3] makes use of Wikipedia as well

[4] https://en.wikipedia.org/wiki/Wikipedia

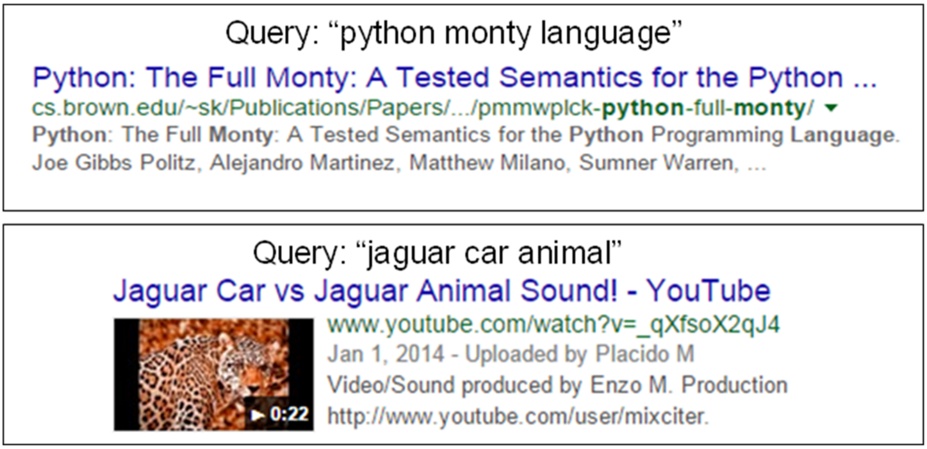
as another knowledge base called ConceptNet in performing diversified query expansion. Apart from BLN, there have been other methods exploiting Wikipedia for the task of query expansion, a well-cited work being [36]. From a query, the technique narrows down to a small subset of Wikipedia pages that are either of (1) top ranked articles from Wikipedia retrieved in response to the query, or (2) the Wikipedia entity pages, in cases where the query is regarded as an entity query, that focused on an entity. Terms are selected from such Wikipedia articles in a pseudo-relevance framework; the authors analyze and evaluate the strategy in addressing query expansion for various categories of queries. It may be particularly noted that, unlike the approaches discussed so far, this work does not address the diversity factor.

2.3.2 Word embeddings

Over the last few years, word embeddings such as word2vec [22] and GloVe [25] have become popular in text processing. These models learn geometric encodings (i.e., vector representations) for words from their co-occurrence information. The methods differ in that word2vec leans a model that can predict a word given a set of 'context' words (or vice versa), whereas GloVe performs dimensionality reduction using co-occurrence information to arrive at vector embeddings. Due to being fairly new, these embeddings are still in the process of being employed for the variety of tasks within information retrieval and search. A recent work [11] proposes the usage of word embeddings in finding a set of related terms to the query term, which is then used to form an expansion language model. This expansion language model is then used to score documents against, completing the retrieval pipeline. Another work [19] proposes scoring candidate query expansion terms using the similarity of their word embeddings to those of the terms in the query. Whole both these methods do not incorporate mechanisms for diversifications within them, we extend the latter model, called *RM-CombSum* with an MMR [5] based diversification, leading to a word-embedding based diversified query expansion method that we will use as a baseline method in our empirical evaluation. The similarity function between terms used in the diversity term is simply the cosine similarity between the corresponding word embedding vectors.

2.4 DQE uptake model

The suggested uptake model for DQE as used in most methods (e.g., [2]) is that the original search query (e.g., *python*) be appended with all the (optionally weighted) terms in the result (e.g., *language, monty*) to form a single large query that is expected to produce a result set encompassing multiple aspects. While this is likely be a good model for search engines that work on a small corpus and other specialized scenarios, we observe that such extended queries are not likely to be of high utility for large-scale search engines. This is so since there is a likelihood of a very rare aspect in the intersection of multiple terms in the extended query that would most likely end up being the focus of the search since search engines do not consider terms as being independent. Figure 1 illustrates a couple of such examples, where very rare and non-noteworthy aspects form part of the top results. Thus, we focus on the model where terms in the DQE result set be separately appended to the initial query to create multiple *aspect-pure* queries. Thus, in our example, we expect that '*python language*' or '*python monty*' be candidates for the user to choose from, in order to expand and re-formulate the initial query, i.e., *python*.



Query: “python monty language”

Python: The Full Monty: A Tested Semantics for the Python ...

cs.brown.edu/~sk/Publications/Papers/.../pmmwplck-**python**-full-**monty**/ ▾

Python: The Full **Monty**: A Tested Semantics for the **Python** Programming **Language**.

Joe Gibbs Politz, Alejandro Martinez, Matthew Milano, Sumner Warren, ...

Query: “jaguar car animal”

Jaguar Car vs Jaguar Animal Sound! - YouTube

▶ 0:22

www.youtube.com/watch?v=_qXfsoX2qJ4

Jan 1, 2014 - Uploaded by Placido M

Video/Sound produced by Enzo M. Production

http://www.youtube.com/user/mixciter.

Figure 1 Sample Results from Extended Queries

3 Problem statement and solution framework

We now outline the problem statement more formally and introduce the solution framework employed by our methods, SLR and SER.

3.1 Problem statement

Given a document corpus $\mathbb{D}$ and a query phrase $\mathcal{Q}$, the *diversified query expansion* (DQE) problem requires that we generate an ordered (i.e., ranked) list of *expansion terms* $\mathbb{E}$. Each of the terms in $\mathbb{E}$ may be appended to $\mathcal{Q}$ to create an extended query phrase that could be processed by a search engine operating over $\mathbb{D}$ using a relevance function such as BM25 [35] or PageRank [23]. The relevance function itself is external to the DQE task. The ideal $\mathbb{E}$ is that ordering of terms such that the separate extended queries formed using the top *few* terms in $\mathbb{E}$ are capable of eliciting documents relevant to *most* aspects of $\mathcal{Q}$ from the search engine. Typically, users are interested in perusing only a few expansion possibilities, with research indicating that as many as 91% of users are unlikely to go beyond the first page of search results in Web search engines [33]; thus, a quality measure for DQE is the aspect coverage achieved over the top-k terms for an appropriate value of k such as 5. *Diversified entity recommendation* (DER) is the analogous problem of generating an ordered list of entities, $\mathcal{E}$, from an ontology (Wikipedia, ConceptNet etc.) such that most diverse aspects of the query are covered among the top few entities. It may be noted that we do not presume availability of usage data (e.g., query logs) or supervision (e.g., documents labelled with aspect relevance information) in addressing the DQE/DER tasks.

3.2 Framework for using semantic resources in diversified query expansion

We now outline our three-phase skeletal framework for diversified query expansion that we base our methods on. The three phases are as follows:

- **Selection:** This phase selects information of relevance to the query from the document corpus used in the retrieval system. Across our methods, we select a subset of terms that are deemed relevant to the query.

- **Correlation:** The information selected in the first phase is now correlated with external semantic resources. We propose separate methods for correlating with Wikipedia and pre-learned word embeddings, as we will illustrate in the next section.
- **Ranking:** This phase involves ranking candidate expansion terms in order to arrive at a final result set, $\mathbb{E}$. In both our methods, we make use of diversity-conscious graph node ranking using the vertex reinforced random walk technique, to rank the expansion terms. However, differences in the previous phase across the methods entail consequent differences in this phase as well.

As outlined earlier, we develop two methods based on this framework, SLR and SER, targeted at using Wikipedia and pre-learned word embeddings respectively. Both the methods are identical in the selection phase, but differ in the subsequent phases. We describe each method in separate sections.

4 Select-Link-Rank: Wikipedia for diversified query expansion

This section describes Select-Link-Rank (SLR), our technique for exploiting Wikipedia for diversified query expansion. Figure 2 outlines the flowchart of SLR. Given a search query, SLR starts by selecting informative terms (i.e., words or tokens) from the results returned by the search engine using a statistical measure. Since we use a large number of search results in the select phase to derive informative terms from, we expect to cover terms related to most aspects of the query. A semantic footprint of these terms is achieved by mapping them to Wikipedia entities in the Link Phase. The sub-graph of Wikipedia encompassing linked entities and their neighbors is then formed. The Rank phase works by performing a diversity-conscious scoring of entities in the entity sub-graph. Specifically, since distinct query aspects are expected to be semantically diverse, the Wikipedia entity sub-graph would likely comprise clusters of entities that roughly map to distinct query aspects. The *vertex-reinforced random walk (VRRW)* ensures that only a few representatives of each cluster,

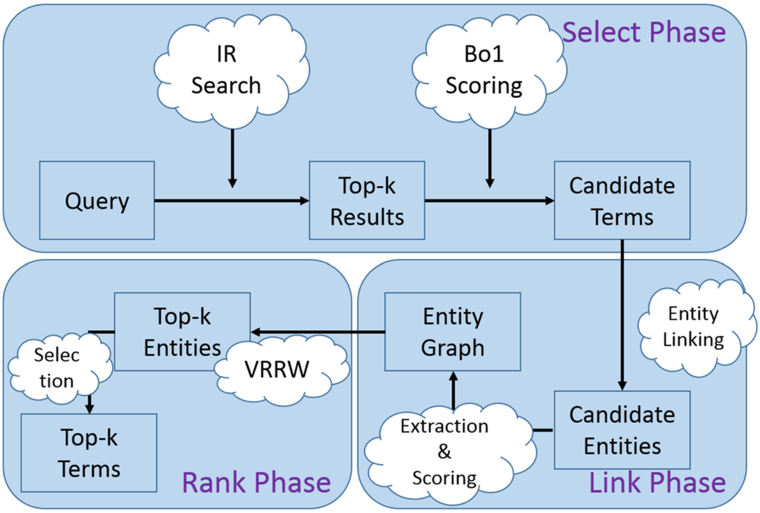


Figure 2 Pipeline of the SLR algorithm

and hence aspect, would get high scores; this produces an aspect-diversified scoring of entities. Such a diversified entity scoring is then transferred to the term space in the last step, achieving a diversified term ranking. The select, link and rank phases correspond to the three phases in the three-phase skeletal framework outlined earlier. In the following sections, we will describe the various phases in SLR. We will use the ambigious query *jaguar* as an example to illustrate the steps in SLR; jaguar has multiple aspects corresponding to many entities bearing the same name. These include an animal species,[5] a luxury car manufacturer,[6] a formula one competitor,[7] a video game console[8] and an American professional football franchise[9] as well as many others.

4.1 Select: Selecting candidate expansion terms

We first start by retrieving the top-K relevant documents to the initial query Q, denoted by $Res_K(Q, \mathbb{D})$ from a search engine operating on $\mathbb{D}$. From those documents, we then choose T terms whose distribution among the top-K documents contrasts well from their distribution across documents in the corpus. This divergence is estimated using the Bo1 model [15], a popular informativeness measure that uses Bose-Einstein statistics to quantify divergence from randomness as below:

$$Bo1(t) = f(t, Res_K(Q, \mathbb{D})) \times log_2 \frac{1 + (f(t, \mathbb{D})/|\mathbb{D}|)}{f(t, \mathbb{D})/|\mathbb{D}|} + log_2(1 + (f(t, \mathbb{D})/|\mathbb{D}|)) \quad (1)$$

where $f(a, B)$ denotes the frequency of the term a in the document collection represented by B. Thus, $f(t, \mathbb{D})/|\mathbb{D}|$ denotes the normalized frequency of t in $\mathbb{D}$. It is notable that Bo1 scoring does not involve any parameter that requires tuning. To ensure all aspects of Q have a representation in $Res_K(Q, \mathbb{D})$, K needs to be set to a large value; we set both K and T to 1000 in our method. The selected candidate terms are denoted as $Cand(Q, \mathbb{D})$. The top Bo1 words for our example query *jaguar* included words such as *panthera* (relating to animal), *cars*, *racing*, *atari* (video game) and *jacksonville* (American football).

Remarks Starting with the top documents from a standard search engine allows our approach to operate as a layer on top of standard search engines. This is important from a practical perspective since disturbing the standard document scoring mechanism within search engines would require addressal of indexing challenges entailed, in order to achieve acceptable response times. Such considerations have made re-ranking of results from a baseline relevance-only scoring mechanism a popular paradigm towards improving retrieval [5, 29].

4.2 Link: Linking to wikipedia and entity graph creation

In this phase, we use the terms in $Cand(Q, \mathbb{D})$ to link to Wikipedia entities leading up to the creation of an entity graph with nodes weighted as a function of their relatedness to the terms. We now outline the steps leading to the creation of the graph in three subsections herein.

[5] https://en.wikipedia.org/wiki/Jaguar

[6] http://www.jaguar.co.uk/

[7] https://en.wikipedia.org/wiki/Jaguar_Racing

[8] http://www.retrogamer.net/profiles/hardware/atari-jaguar-2/

[9] http://www.jaguars.com/

4.2.1 Identifying relevant wikipedia entities

We link each term in $Cand(Q, \mathbb{D})$ to one or more related Wikipedia entities that are deemed to be relevant to the term. Since our candidate terms are targeted towards extending the original query, we form an extended query for each candidate term by appending the term to Q. We then leverage entity linking methods, such as TagMe [13] and [10], which match small text fragments with entity descriptions in Wikipedia to identify top-related entities. It may be noted that the specific method employed for entity linking can be substituted with better methods that may become available with advances in the field.

Thus, eventually, each term t in $Cand(Q, \mathbb{D})$ is associated with a set of entities, $t.E$. Typical entity linking methods, in addition to identifying relevant entities to link to, are also able to quantify the relatedness between the text fragment and the entity. We use $r(t, e)$ to denote the relatedness score between term t and entity e (in $t.E$) as estimated by the entity linking technique. In case entity linking methods that do not quantify the strength are employed, the corresponding $r(t, e)$ would simply be set to unity.

For our example, *panthera* got linked to the *Jaguar* and *Panthera* entities whereas *cars* brought in entities such as *Jaguar Cars* and *Jaguar E-type*. The *racing* related entities were *Jaguar Racing* and *Tom Walkinshaw Racing*. Jaguar E-type was observed to be a type of Jaguar car, whereas Tom Walkinshaw Racing is an auto-racing team very closely associated with Jaguar Racing.

4.2.2 Wikipedia subgraph creation

We now use the information from entity linking to form an entity graph. Our entity graph is a subgraph of the Wikipedia entity graph; the Wikipedia entity graph is simply the set of entities in Wikipedia as nodes, with each hyperlink from an entity article corresponding to entity e to the entity article corresponding to e' translating to an unweighted edge $e \rightarrow e'$. We now describe the construction of our entity subgraph of the Wikipedia graph, which we denote as $G(\mathcal{Q}) = \{V(\mathcal{Q}), E(\mathcal{Q})\}$. Informally, $V(\mathcal{Q})$ comprises all entities that are directly linked to a term in $Cand(Q, \mathbb{D})$ or is a neighbor of such a term; the set of edges $E(\mathcal{Q})$ is then the subset of Wikipedia graph edges connecting entities within $V(\mathcal{Q})$. More specifically, $V(\mathcal{Q}) = N_1 \cup N_2$ where

$$N_1 = \{\cup_{t \in Cand(Q,\mathbb{D})} t.E\} \tag{2}$$

$$N_2 = \{e \mid \exists e' \in N_1,\ e \notin N_1,\ (e', e) \in E_W\} \tag{3}$$

where E_W is the set of all links in the Wikipedia Graph. The edge set $E(\mathcal{Q})$ has representation from all Wikipedia links between nodes in $V(\mathcal{Q})$. Here, N_1 captures entities linked to candidate terms. N_2 brings in their one-hop outward neighbors not already covered by N_1. In other words, N_2 contains entities that are directly related to the linked entities and could therefore enrich our understanding of the aspects related to the query. The inclusion of one-hop neighbors, while being a natural first step towards expanding the concept graph, is related to the inclusion of all nodes along two-hop paths between nodes in N_1; the latter heuristic has been used in knowledge graph expansion in [28]. For the *jaguar* example, N_2 was seen to comprise entities such as *Formula One* that was found to connect to both *Jaguar Racing* and *Jaguar Cars* entities, thus uncovering the connection between their respective aspects.

4.2.3 Entity importance weights

Having built the graph $G(\mathcal{Q})$, we now assign entity importance weights to nodes in $V(\mathcal{Q})$ leveraging information about its relatedness to terms in $Cand(Q, \mathbb{D})$ and it's connectedness to other nodes in the graph. We start with assigning weights to entities that are directly linked to terms in $Cand(Q, \mathbb{D})$:

$$wt'(e \in N_1) = \frac{\sum_{t \in Cand(Q,\mathbb{D})} I(e \in t.E) \times r(t, e)}{\sum_{e' \in N_1} \sum_{t \in Cand(Q,\mathbb{D})} I(e' \in t.E) \times r(t, e')} \tag{4}$$

where $I(.)$ is the identity function. Thus, the weight of each entity in N_1 is set to be the sum of the relatedness scores from each term that links to it. This is normalized by the sum of weights across entities in N_1 to yield a distribution that sums to 1.0. The weights for those in N_2 uses the weights of N_1 and is defined as follows:

$$wt'(e \in N_2) = \frac{max\{wt(e')|e' \in N_1,\ (e', e) \in E(\mathcal{Q})\}}{\sum_{e'' \in N_2} max\{wt(e')|e' \in N_1,\ (e', e'') \in E(\mathcal{Q})\}} \tag{5}$$

Thus, the weight of nodes in N_2 is set to that of their highest scored inward neighbor in N_1, followed by normalization. The other option, using *sum* instead of *max*, could cause some highly connected nodes in N_2 to have much higher weights than those in N_1. In the interest of arriving at an importance probability distribution over all nodes in $G(\mathcal{Q})$, we do the following transformation to estimate the final weights:

$$wt(e) = \begin{cases} \alpha \times wt'(e) & e \in N_1 \\ (1 - \alpha) \times wt'(e) & e \in N_2 \end{cases} \tag{6}$$

where $\alpha \in [0, 1]$ is a parameter that determines the relative importance between directly linked entities and their one-hop neighbors. Intuitively, this would be set to a high value to ensure directly linked entities have higher weights than one-hop neighbors. This completes the graph construction and thus the Link phase of SLR.

4.3 Rank: Ranking candidate terms

This phase uses the graph $G(\mathcal{Q})$ and associated node-importance weights to arrive at a the final DQE result set, i.e., an ordered list of terms, $\mathbb{E}$. We model this phase as two sub-phases, the first that scores entities in $G(\mathcal{Q})$ in diversity-conscious fashion and the second that translates such scoring to the space of terms.

4.3.1 Vertex reinforced random walk

Our goal here is to rank the linked entities based on their diversity and relevance. For that purpose, the nodes in $G(\mathcal{Q})$ are scored using a diversity-conscious adaptation of PageRank [23] that does a *vertex reinforced random walk (VRRW)* [24]. VRRW is similar to PageRank, but it is a time-variant *random walk* process. A random walk on a network defines a *Markov chain*, where each node represents a state and a walk transits from node u to node v proportional to the transition probability, denoted as $p(u, v)$. Transitions happen only through edges in the network and the transition probabilities determine the next node to visit. While in PageRank the transition probability $p(e, e')$ between any two nodes $e,\ e'$ is static, in VRRW, the transition probability to a node (entity) e' is reinforced by the number of previous visits to e'. The impact of this reinforcement can be seen in Figure 3, wherein the final node weights are redistributed to a more mutually diverse set of nodes.

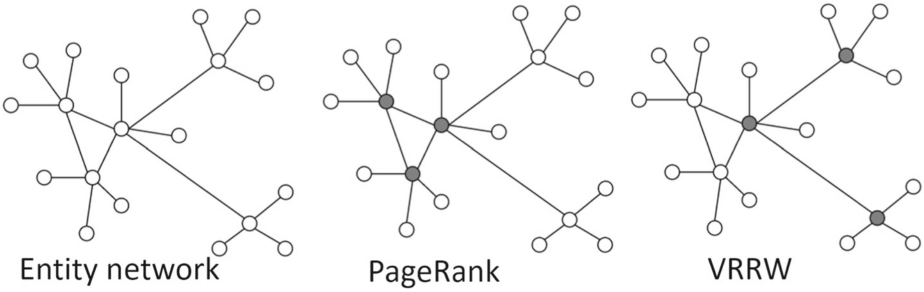


Figure 3 The three nodes (shaded) with the highest scores in PageRank vis-a-vis VRRW

Once the VRRW is started, it proceeds by generating a random number $r \in [0, 1]$ at each iteration, and using it along with the transition probability to choose the next node to visit. To formalize VRRW, let $p_0(e, e')$ be the transition probability from e to e' at timestamp 0, which is the start of the random walk. In our problem, $p_0(e, e') \propto wt(e')$. Now, let $N_T(v)$ be the number of times the walk has visited e' up to time T. Then, VRRW is defined sequentially as follows. Initially, $\forall e \in V(\mathcal{Q}),\ N_0(e) = 1$. Suppose the random walker is at node e at the current time T. Then, at time $T + 1$, the random walk moves to some node e' with probability $p_T(e, e) \propto p_0(e, e')N_T(e')$. Furthermore, for each node in $V(\mathcal{Q})$, we also add a self edge. VRRW is therefore generalized as follows.

$$p_T(e, e') = \lambda\ wt(e') + (1 - \lambda)\frac{wt(e')N_T(e')}{D_T(e)} \tag{7}$$

where $D_T(e) = \sum_{(e,e')\in E(\mathcal{Q})} wt(e')N_T(v)$ is the normalizing term. Here, λ is the teleportation probability, which is also present in PageRank. $(1 - \lambda)$ represents the probability of choosing one of the neighboring nodes based on the reinforced transition probability. However, with probability λ the random walk chooses to restart from a random node based on the initial scores of the nodes. If the network is ergodic, VRRW converges to some stationary distribution of scores over nodes, denoted as $S(\cdot)$, after a large T, i.e., $S(e') = \sum_{e\in V(\mathcal{Q})} p_T(e, e')S(e)$ [24]. Furthermore, $\sum_{\forall e\in V(\mathcal{Q})} S(e) = 1$. The higher the value of $S(e)$ of an entity e, the more important e is. *The top scored entities (nodes) at the end of this phase, $\mathcal{E}$, form the entity recommendation (DER) output of SLR.* The top-5 entities for our example query were found to be: *Jaguar Cars*, *Jaguar* (the entity corresponding to the animal species), *Atari Jaguar* (video game), *Jaguar Racing* and *Jacksonville Jaguars*. The next section describes how this entity scoring can be transferred to the term space to form the DQE output.

Why does VRRW favor representativeness? It is useful to consider how VRRW favors representativeness despite the formulation being very similar to PageRank. As in PageRank, nodes with higher centralities get higher weights due to the flow arriving at these nodes. This, in turn results in larger visit counts $(N_T(v))$. When the random walk proceeds, the nodes that already have high visit counts tend to get an even higher weight. In other words, a high-weighted node starts dominating all other nodes in its neighborhood; such vertex reinforcement induces a competition between nodes in a highly connected cluster leading to an emergence of a few clear leaders per cluster as illustrated in Figure 3.

4.3.2 Diversified term ranking

The DQE output, $\mathbb{E}$, is now constructed using the entity scores in $S(.)$. In the process of constructing $\mathbb{E}$, we maintain a set of entities that have already been *covered* by terms already chosen in $\mathbb{E}$ as $\mathbb{E}.E$. An entity is said to be covered if a term that it was considered relevant to (Refer Section 4.2.1), has already been chosen in the growing set $\mathbb{E}$. At each step, the next term to be added to $\mathbb{E}$ is chosen as follows:

$$t^* = \underset{t \in Cand(\mathcal{Q},\mathbb{D})}{\operatorname{argmax}} \sum_{e \in t.E} I(e \notin \mathbb{E}.E) \times r(t, e) \times S(e) \tag{8}$$

Informally, we choose terms based on the sum of the scores of linked entities weighted by relatedness (i.e., $r(t, e)$), while excluding entities that have been *covered* by terms already in $\mathbb{E}$ to ensure diversification. The generation of $\mathbb{E}$, the DQE output, completes the SLR pipeline. The top-5 expansion terms for the *jaguar* query were found to be: *car*, *onca,*[10] *atari*, *jacksonville*, *racing*. It is notable that despite *cars* and *racing* aspects being most popular on the Web, other aspects are prioritized higher than *racing* when it comes to expansion terms. This is so due to the presence of entities such as *Formula One* in the entity neighborhood (i.e., N_2) that uncover the latent connection between the *racing* and *cars* aspects; VRRW accordingly uses the diversity criterion to attend to other aspects after choosing *cars*, before coming back to the related *racing* aspect.

4.4 Computational costs

We briefly analyze the computational efficiency of SLR, in the interest of understanding its scalability.

- The **Select** phase makes use of a search engine such as *Indri* [31] to run the queries, which might internally make use of language modelling and inference networks to perform the search. The system is reported to be quite fast delivering response times of the order of a second, as outlined in [31]. Selection of T terms from K retrieved documents can be performed using a heap, at a cost of $\mathcal{O}(K \times L_{max} + W_u \times log(K'))$ where L_{max} is the maximum number of non-stop-words per document, and W_u is the total number of unique words.
- In the **Link** phase, each of the T chosen terms from the previous phase are used to expand queries and link to entities. This is performed using a reverse index from words to Wiki pages and a scoring mechanism such as TF-IDF.[11] Computational costs depend on the number of candidate pages, which is roughly proportional to the total number of pages (with a very small constant), and inversely to the vocabulary of the corpus (number of unique words).
- The **Rank** phase involves VRRW, whose matrix implementation takes time quadratic in $\mathcal{O}(|S|^2)$ per iteration, where $|S|$ denotes the number of nodes in S, the graph over which VRRW is executed. In practice, we found VRRW to converge in less than 15 iterations, leading to very fast computations in the order of a few seconds.

The main target of optimization for resource constrained scenarios such as systems that expect real-time responses would be the *Rank* phase, being the only phase that has quadratic

[10] P. Onca is the scientific name of the wild cat called Jaguar

[11] https://en.wikipedia.org/wiki/Tf-idf

complexity. However it is possible to find an efficient tradeoff between the number of candidate expansion terms considered and the computation time.

4.5 Summary and remarks

The various steps in SLR and their sequence of operation are outlined in the pseudocode in Algorithm 2. Since the separate phases have been covered in detail in the previous section, we do not explain them further. It may be noted that we do not make use of wikipedia disambiguation pages in SLR. While wikipedia disambiguation pages are useful, they are generally available only for topics of broad-based interest, and a technique relying on them would not be applicable for queries focused on niche entities. Further, this ensures fairness in comparison with the baselines that do not use curated disambiguations.

Algorithm 1 SLR: Select-Link-Rank

```
Input: Query $\mathcal{Q}$, corpus $\mathbb{D}$, $Wikipediagraph$
Output: List of diversified expansion terms, $\mathbb{E}$, and diversified entities, $\mathcal{E}$
Select Phase
1. Retrieve $K$ result documents for search query $\mathcal{Q}$
2. Select $T$ informative terms from them as $Cand(\mathcal{Q}, \mathbb{D})$
Link Phase
3. Link each term $t$ in $Cand(\mathcal{Q}, \mathbb{D})$ to Wikipedia
4. Let linked entities be $t.E$ and relatedness score be $r(t, e)$
5. Construct $G(\mathcal{Q})$, graph of linked entities and neighbors
6. Score each entity using relatedness to linked terms
Rank Phase
7. Perform VRRW on $G(\mathcal{Q})$, entity scores initialized using (6)
8. Collect the top-scored entities based on VRRW scores as $\mathcal{E}$
9. Construct $\mathbb{E}$, a diversified term ranking using entity scores and term-entity relatedness.
```

5 Select-Embed-Rank: Word embeddings for diversified query expansion

We now outline our approach targeted at exploiting a word embeddings, another semantic resource that has gained much recent popularity, for the task of diversified query expansion. Word embeddings are word-specific vectors learnt by making use of word co-occurrence information. Unlike Wikipedia which is an encyclopaedic semantic resource, word embeddings can be generated even for specialized corpora. For example, word embeddings learnt from a corpus of medical documents would be able to characterize the semantics in the medical domain better than by usage of a generic resource like Wikipedia. This wider reach that an embedding-based DQE method would have motivates the need for a method that can exploit word embeddings in diversified query expansion. In this paper, we restrict our empirical evaluation to a generic search setting, so that SER may be compared against SLR on a fair footing. Figure 4 outlines the flow of the SER technique. The select phase is identical to that of SLR, and involves selecting top informative terms from the search results. This is followed by the Embed phase where the corresponding word embeddings are fetched from a dataset of pre-learned word embeddings. The similarities between the terms are estimated using a similarity measure between the corresponding word embedding vectors. These similarities are used in creating a term graph, which form the input to the Rank phase. In contrast

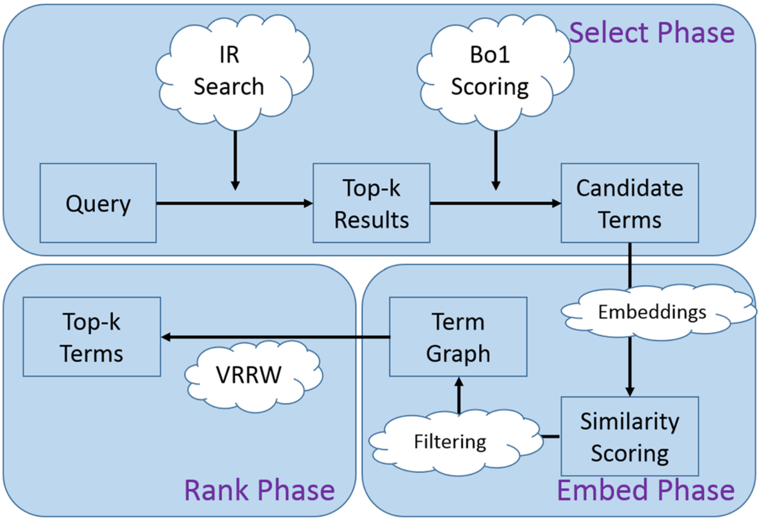


Figure 4 Pipeline of the SER algorithm

to SLR, the SER rank phase involves using the VRRW walk directly on the term graph, resulting in a term scoring that forms the DQE output. We will now describe the various phases in detail in subsections herein.

5.1 Select: Selecting candidate expansion terms

The select phase in SER is identical to the select phase in SLR as outlined in Section 4.1. It involves selecting the top-T terms from across the top-K documents retrieved in response to the search query, Q. The Bo1 measure is used to score terms, resulting in a candidate set $Cand(Q, \mathbb{D})$. Due to the usage of the initial result set from a relevance-only search, SER is also amenable to be used within an IR re-ranking framework.

5.2 Embed: Using word embeddings for term graph construction

The embed phase brings in word embeddings into the picture. SER was designed in order to be able to leverage pre-learned word embeddings such as the Google News word2vec[12] or the Wikipedia/Twitter/Gigaword GloVe vectors[13] in diversified query expansion. While we will consistently make use of such pre-trained vectors in our empirical evaluation, the framework itself only expects to be able to map each term from $Cand(Q, \mathbb{D})$ to a vector; thus, for very specialized-domain search systems, it would be appropriate to use word vectors learnt from the corpus $\mathbb{D}$ itself. This phase involves the construction of an initial term graph using word embedding similarities, followed by refining it by heuristically filtering out edges and vertices.

[12] https://code.google.com/archive/p/word2vec/

[13] http://nlp.stanford.edu/projects/glove/

5.2.1 Term graph construction

Let the word embeddings for a term t be represented by $t.V$; the word embedding is a numeric vector of fixed dimensionality, usually $100 - 300$. We now construct a graph $G^0(\mathcal{Q}) = \{V^0(\mathcal{Q}), E^0(\mathcal{Q})\}$. $V^0(\mathcal{Q})$ simply comprises all terms in $Cand(Q, \mathbb{D})$. The edge set is defined as follows:

$$E^0(\mathcal{Q}) = \{(t, t', sim(t.V, t'.V)) | \{t, t'\} \subseteq V^0(\mathcal{Q}) \wedge t \neq t' \wedge sim(t.V, t'.V) \geq \tau\} \quad (9)$$

with the triplet (t, t', s) denoting that there would be a directed edge from t to t' bearing a weight s. Thus, we induce an edge between any two terms if a measure of similarity between their corresponding embedding vectors, defined as $sim(.,.)$, exceeds a threshold τ. Since we do not impose any constraint on $sim(.,.)$, any similarity measure that quantifies similarity in [0, 1] could be used; we consistently use cosine similarity, being a popular similarity measure for numeric vector data.

5.2.2 Term graph refinement

We employ some general heuristics to now refine the graph $G^0(\mathcal{Q})$ by filtering out nodes and edges, in order to arrive at our final term graph $G(\mathcal{Q}) = \{V(\mathcal{Q}), E(\mathcal{Q})\}$. We separately outline the intuition and operation of each of our heuristics herein.

General Word Filtering Heuristic The distributional assumption involved in learning the word embeddings attempts to build word vectors that are good at explaining the context in which the word appears in the corpus. This causes words denoting different instances of the same type to map to similar vectors. As an example, we observed that the vector for the word *washington* bears high similarity to words such as *iowa*, *michigan* and *mumbai* since place names appear within similar contexts. To further outline how it could affect query expansion, let us consider the query *jennifer actress* that is meant to focus on actresses with a forename *jennifer*. The aforementioned nature of word embeddings causes words that relate to other actresses, regardless of their forenames, to be highly connected to terms related to the query thus exaggerating their importance in the diversified query expansion process. To avoid this, our general word filtering heuristic filters the node set as follows:

$$V(\mathcal{Q}) = \{t | t \in V^0(\mathcal{Q}) \wedge |Neighbors(t, E^0(\mathcal{Q}))| \leq |V^0(\mathcal{Q})| \times \mu\%\} \quad (10)$$

where $Neighbors(t, E^0(\mathcal{Q}))$ denotes the set of nodes that are connected to t through edges in $E^0(\mathcal{Q})$. Thus, all nodes in $V^0(\mathcal{Q})$ that are linked to more than $\mu\%$ terms under $G^0(\mathcal{Q})$ would be eliminated leading to a refined set of nodes. This heuristic is related to and inspired by the sampling strategy used in [20].

Edge Limit Heuristic Frequently occurring terms within $Cand(Q, \mathbb{D})$ would typically be placed in dense neighborhoods in the embedding space due to their co-occurrence with a a large variety of terms. Consequently, they would be very highly connected in the $G^0(\mathcal{Q})$ graph, and could exert high influence in the graph traversal that we will employ in the Rank phase. *movie* is an example of such a term for the query *jennifer actress* that is highly connected due to this property. In order to limit the influence of such common terms, we limit the maximum number of edges that can originate from a node in the term graph by choosing the top-ρ edges with the highest weights. This leads to the following filtering:

$$E(\mathcal{Q}) = \{(t, t', s) | (t, t', s) \in E^0(\mathcal{Q}) \wedge t' \in Top\text{-}\rho(t, E^0(\mathcal{Q}))\} \quad (11)$$

where $Top\text{-}\rho(t, E^0(\mathcal{Q}))$ denotes the top-ρ edges originating from t within $E^0(\mathcal{Q})$ when the edges are sorted based on their scores.

Applying the above two heuristics to filter the graph $G^0(\mathcal{Q})$ leads to the refined graph $G(\mathcal{Q})$ that will be used in the next step.

5.3 Rank: Ranking candidate terms

This phase, much like the analogous phase in SLR, employs a VRRW to score terms in the graph in a diversity-conscious fashion. Unlike the SLR version, we do not use node importance weights in the SER graph, and thus, the transition probability is uniform across all the edges. An important distinction between SLR and SER being that the former employs VRRW on the entity graph whereas we use VRRW on the term graph directly. Once the VRRW stabilizes, we are left with a score for each term in the term graph, which we denote as $S(t)$. The DQE output, $\mathbb{E}$ is then the set of terms in $V(\mathcal{Q})$ ordered in the decreasing (or non-increasing, to be precise) order of the scores according to $S(.)$.

5.4 Using SER for diversified entity recommendation

Due to the non-usage of an entity knowledge base such as Wikipedia or ConceptNet within the SER pipeline, the DQE output $\mathbb{E}$ needs to be adapted in conjunction with an entity knowledge base to form a diversified entity ranking output, $\mathcal{E}$, for usage as a DER method. We accomplish this using a suitable entity linking method, such as those was discussed in Section 4.2.1. Specifically, for each term in the $\mathbb{E}$ output, we append the query with that term forming a text segment, which would then be used in an entity linking system to choose the most related entity as the following:

$$t.entity = \arg\max_{e \in t.E} \; r(t, e) \tag{12}$$

where $t.E$ is the set of entities linked to the text segment formed by collating the query with the term t and $r(t, e)$ is a relationship strength output by the linking method (all notations same as in Section 4.2.1).

Algorithm 2 SER: Select-Embed-Rank

```
Input: Query $\mathcal{Q}$, corpus $\mathbb{D}$, pre-learned word embeddings
Output: List of diversified expansion terms, $\mathbb{E}$, and diversified entities, $\mathcal{E}$
Select Phase
1. Retrieve $K$ result documents for search query $\mathcal{Q}$
2. Select $T$ informative terms from them as $Cand(\mathcal{Q}, \mathbb{D})$
Embed Phase
3. Retrieve the word embedding vector corresponding to each $t \in Cand(\mathcal{Q}, \mathbb{D})$ as $t.V$
4. Form a graph with nodes as terms from $Cand(\mathcal{Q}, \mathbb{D})$
5. Build an edge between two terms if the similarity between their embedding vectors
exceed $\tau$
6. Refine the term graph by filtering edges and nodes using heuristics outlined in Section
5.2.2
Rank Phase
7. Perform VRRW on the refined term graph $G(\mathcal{Q})$
8. Collect the top-scored terms based on VRRW scores as $\mathbb{E}$
9. Replace each term in $\mathbb{E}$ by the most similar entity in a knowledge base as in Eq. 12,
   to form the DER output $\mathcal{E}$
```

5.5 Computational costs

Similar to Section 4.4, we analyse the computational costs of the various phases in SER.

- The **Select** phase in SER, being identical to SLR, involves invocation of an IR engine such as Indri [31].Selection of T terms from K retrieved documents involves a cost of $\mathcal{O}(K \times L_{max} + W_u \times log(K'))$ where L_{max} is the maximum number of non-stop-words per document, and W_u is the total number of unique words.
- The SER **Embed** phase differs significantly from the SLR *Link* phase in that it involves building a graph spanning the T terms selected in the previous phase. In the absence of any indexes, the graph construction is $\mathcal{O}(T^2)$. However this can be completely offset by maintaining a pre-computed index of similar terms which would result in a linear complexity of $\mathcal{O}(\rho T)$, ρ being the edge-limit.
- The **Rank** phase, being identical to SLR, is $\mathcal{O}(|S|^2)$ where $|S|$ denotes the size of the refined term graph.

In summary, SER is seen to be quadratic in the *Rank* phase. However, since the *Rank* phase graph has fewer nodes than the initial *Embed* phase graph, the *Embed* phase could be prioritized for optimization by way of usage of a pre-computed similarity index over the distributional word embeddings.

5.6 Summary

Algorithm 2 illustrates the various steps in the SER method in a pseudocode. As indicated in previous sections, the major difference between the SLR and SER is in the Correlation phase in the three-phase framework (Section 3.2), where different strategies are adopted to make use of respective semantic resources, motivated by the nature of their different characteristics.

6 Experiments

6.1 Experimental setup

We use the ClueWeb09 [7] Category B dataset comprising 50 million Web pages in our experiments. In SLR and SER, we use the publicly accessible Indri interactive search interface for procuring initial results. This was followed by usage of a simple custom entity linker based on Apache Lucene [17]; specifically, all entities were indexed using their article body text, and the top-result entities in response to each term were used as linked entities along with their corresponding relevance scores. We now detail the default parameter settings for our methods. For the Select phase parameters, we set $K = K' = 1000$ across both the methods. The SLR link phase parameter α is set to 0.65 and we set $\lambda = 0.2$. Meanwhile, the SER embed phase parameters are set as $\tau = 0.4$, $\mu = 4$ and $\rho = 5$. The VRRW restart probability in the Rank phase of both the methods was set to 0.25. We consistently use a query set of 15 queries gathered across motivating examples in papers on SRD and DQE.

We compare our DQE results against LDA-based ts_{xQuAD} [34] where we set the #topics to 5. SLR's DER results are compared against that of BHN [2]. For both ts_{xQuAD} and BHN, all parameters are set to values recommended in the respective papers.

We use both user studies and automatic evaluations in order to assess the empirical performance of our methods, SLR and SER. With the limited amount of resources available for the user study, we choose to do two sets of user evaluations; (i) benchmarking SLR on both DQE and DER against respective baselines, and (ii) evaluating SER against SLR on the DQE task. The user study was rolled out to an audience of up to 100 technical people (grad students and researchers) of whom around 50% responded. The users were free to choose one or more of the four surveys to respond to, thus leading to different numbers of votes for each of the four surveys. All questions were optional; thus, some users only entered responses to a few of the queries even within a survey. Since the user study was intended to collect responses at the result-set level to reduce the number of entries in the feedback form, we are unable to use evaluation measures such as α-NDCG that require relevance judgements at the level of each result-aspect combination. Apart from the user study, we also perform an automated diversity evaluation focused on the DQE task.

SER Variants SER is designed to be able to make use to pre-learned word embeddings. In the interest of evaluating its performance over various word embeddings, we instantiate SER with two different sets of pre-learned word embeddings. The first set is that of GloVe embeddings trained on the Wikipedia dataset[14] and the second set is the set of word2vec embeddings trained on Google News.[15] We refer to these as SER-Wiki and SER-News respectively. While SER-Wiki is expected to perform better due to the generality of the Wikipedia dataset, the performance of the Google News embeddings would indicate the suitability of using word embeddings from domains that are slightly divergent to the text corpus used in the retrieval system.

6.2 User study results

For each user study, two methods are pitched against each other. For each of the 15 queries in our query set, we generate the top-5 results (terms for the DQE task and entities for the DER task) by both the methods and request users to choose the better result set. The survey itself was randomized; thus, for one query, results from the first method could appear on the left, while it might be on the right for another query.

6.2.1 DQE evaluation: SLR vs. ts_{xQuAD}

The vote distribution for this study is illustrated in the left half of Table 2. SLR is seen to be preferred over ts_{xQuAD} across all queries, with the preference being strongest for queries such as *java* (41-2) followed by *fifa 2006*, *rock and roll* and *jennifer actress*. 87% of user inputs were seen to favor SLR, thus indicating a strong preference for SLR expansion suggestions.

6.2.2 DER evaluation: SLR vs. BHN

Results from the results of the DER task benchmarking SLR against BHN appear in the right half of Table 2. The vote distribution suggests that users strongly prefer SLR over BHN on 14 queries while being ambivalent about the query "python". Our analysis revealed that BHN had entities focused on the reptile and the programming language, while our method

[14] http://nlp.stanford.edu/projects/glove/

[15] https://code.google.com/archive/p/word2vec/

Table 2 #Votes from User Study: Expansions (SLR vs. $\mathbf{ts_{xQuAD}}$) & Entities (SLR vs. BHN)

Query Information		DQE Expansions Eval.		DER Entities Eval.	
Sl#	Query	SLR	$\mathbf{ts_{xQuAD}}$	SLR	BHN
1	coke	**37**	6	**40**	11
2	fifa 2006	**40**	3	**33**	18
3	batman	**32**	11	**49**	2
4	jennifer actress	**40**	3	**48**	3
5	phoenix	**39**	4	**42**	10
6	valve	**38**	5	**40**	12
7	rock and roll	**40**	3	**46**	4
8	amazon	**39**	4	**39**	13
9	washington	**37**	6	**38**	12
10	jaguar	**37**	6	**46**	5
11	apple	**30**	14	**41**	9
12	world cup	**36**	8	**50**	1
13	michael jordan	**39**	4	**36**	13
14	java	**41**	2	**41**	9
15	python	**39**	4	25	**26**
	Average	**37.6**	5.53	**40.9**	9.87
	Percentage	**87%**	13%	**81%**	19%

also had results pertaining to a British comedy group, *Monty Python*; we suspect most users were unaware of that aspect for python, and thus did not credit SLR for considering that.

6.2.3 DQE evaluation: SLR vs. SER-Wiki and SLR vs. SER-News

Table 3 lists the vote distribution for the two pairs of user study conducted, with the left half representing the information from the SLR vs. SER-Wiki study and the right half comprising results from SLR vs. SER-News. In both the surveys, SLR was seen to be able to provide better query expansions, with the rich semantic structure of the Wikipedia graph at its disposal. The relative performance of SER-Wiki and SER-News against SLR also agree to expected trends; the more general Wiki embeddings were seen to be useful in prioritizing expansion terms better, whereas the embeddings learnt from the News corpus were judged to be of slightly lesser quality. As an example of how the divergence in character between the embedding datasets reflect in the expansion results, let us consider the query *amazon* from our evaluation dataset. The top-5 terms from SER-Wiki were *river, book, tv, album* and *environmental*. On the other hand, those from SER-News were found to be *music, love, software, book* and *increase*. It is notable that SER-News does not have even one term relating to the river aspect of the query among the top-5, with all terms relating to the company aspect. This is on expected lines given the dominance of the company aspect in news articles.

While we did not perform a direct comparison between the DQE results of the SER versions against those from ts_{xQuAD} to limit the amount of user effort to be requested to within reasonable limits,[16] it is of interest to compare the SER results with those from

[16] The Table 3 surveys which were administered after those for Table 2 already show much lesser participation, indicating that user enthusiasm in survey participation was declining rapidly; this led us to decide against administering a third set of surveys.

Table 3 #Votes from DQE User Study: [SLR vs. SER-Wiki] & [SLR vs. SER-News]

Query Information		User Study Results			
Sl#	Query	SLR	SER-Wiki	SLR	SER-News
1	coke	13	**14**	**12**	8
2	fifa 2006	**14**	13	9	**11**
3	batman	6	**21**	**14**	6
4	jennifer actress	**20**	7	**13**	7
5	phoenix	**17**	10	**10**	**10**
6	valve	**24**	3	**18**	2
7	rock and roll	**18**	9	7	**13**
8	amazon	**15**	12	**15**	5
9	washington	**21**	6	**17**	3
10	jaguar	**17**	10	**15**	5
11	apple	13	**14**	**11**	9
12	world cup	**22**	5	**18**	2
13	michael jordan	**23**	4	**17**	3
14	java	6	**21**	**16**	4
15	python	**16**	11	**12**	8
	Average	**16.33**	10.67	**13.6**	6.4
	Percentage	**60%**	40%	**68%**	32%

Table 2 to draw indicative conclusions about the likely relative performance of SER against ts_{xQuAD}. In the comparison with SLR, ts_{xQuAD} was judged favorably in 13% of the user inputs. On the other hand, SER-Wiki and SER-News were judged favorably in 40% and 32% of user inputs respectively. While these numbers cannot be directly compared against each other due to them being from separate studies against SLR, these do indicate that SER-Wiki and SER-News are likely to perform better than ts_{xQuAD}.

6.3 Automated diversity evaluation

We further evaluate the performance of our methods with respect to the diversity of the aspects represented by the expansion terms and their relevance. Since all previous efforts on DQE use evaluation measures that are based on expensive human-inputs in the form of releveance judgements (e.g., [4, 27]), we now devise an intuitive and automated metric to evaluate the diversity of DQE results by mapping them to the entity space where external entity relatedness measures can be exploited. In other words, this evaluation measure quantifies the diversity of the entities that the DQE output maps to. This allows us to compare all our three methods, SLR, SER-Wiki and SER-News against the baseline methods ts_{xQuAD}, *RM-CombSum-Wiki* and *RM-CombSum-News*. The last two methods are the MMR-based extensions of the method from citekuzi over the Wiki and Google News word embeddings respectively. Consider the top-k query expansions as $\mathbb{E}$; we start by finding the set of entity nodes associated with those expansions, $\mathbb{N}$. We then define an entity-node relevance score $r_{\mathbb{E}}(n)$ as the sum of its relevance scores across its associated expansion terms; i.e., $r_{\mathbb{E}}(n) = \sum_{t \in \mathbb{E}} r(t, n)$. Let $S(n_i, n_j)$ denote an entity-pair semantic relatedness estimate from an external oracle; our quality measure is:

$$Q(\mathbb{E}, \mathbb{N}) = \frac{1}{\binom{|\mathbb{N}|}{2}} \sum_{(n_i, n_j) \in \mathbb{N}} r_{\mathbb{E}}(n_i) \times r_{\mathbb{E}}(n_j) \times exp(-S(n_i, n_j)) \tag{13}$$

where $exp(-S(n_i, n_j))$, as the formula suggests, is a positive value inversely related to similarity between the corresponding entities. Intuitively, it is good to have highly relevant entities to be less related to ensure that entity-nodes in $\mathbb{N}$ are diverse. Thus, *higher values* of the $Q(.,.)$ metric are desirable. We use two versions of Q by separately plugging in two different estimates of semantic similarity to stand for the oracle:

$$S_J(n_i, n_j) = \frac{n_i.neighbors \cap n_j.neighbors}{n_i.neighbors \cup n_j.neighbors} \tag{14}$$

$$S_D(n_i, n_j) = Dexter(n_i, n_j) \tag{15}$$

where $n.neighbors$ indicate the neighbors of the node n according to the Wikipedia graph, and $Dexter(.,.)$ denotes the semantic similarity from Dexter [6].

Figures 5 and 6 show the expansion qualities based on Jaccard and Dexter respectively for the SLR, SER-Wiki, SER-News, ts_{xQuAD}, *RM-CombSum-Wiki* and *RM-CombSum-News* methods. It may be noted that the values are plotted in log-scale to allow for better visualization since the techniques vary much in terms of the evaluation measure; the quality measure being in [0, 1], the log-scale yields all negative values with all the bars in the figure seen to be 'hanging' from the x-axis rather than being held upright. Since higher values (i.e., smaller negative values) are desirable, shorter (hanging) bars correspond to better performance. On an average, across all queries, SLR was seen to outperform all the other methods on both the evaluation measures. SER-Wiki comes next convincingly beating the other methods. Though SER-News was seen to be slightly better than ts_{xQuAD} and *RM-CombSUM-News*, the difference in the quality measure was less than an order of magnitude on an average; note that, due to the log-scale plot, each unit of "length" corresponds to significant deterioration in the quality metric. The main high-level observation from the automated diversity evaluation is that our methods SLR and SER-Wiki significantly outperform the baseline

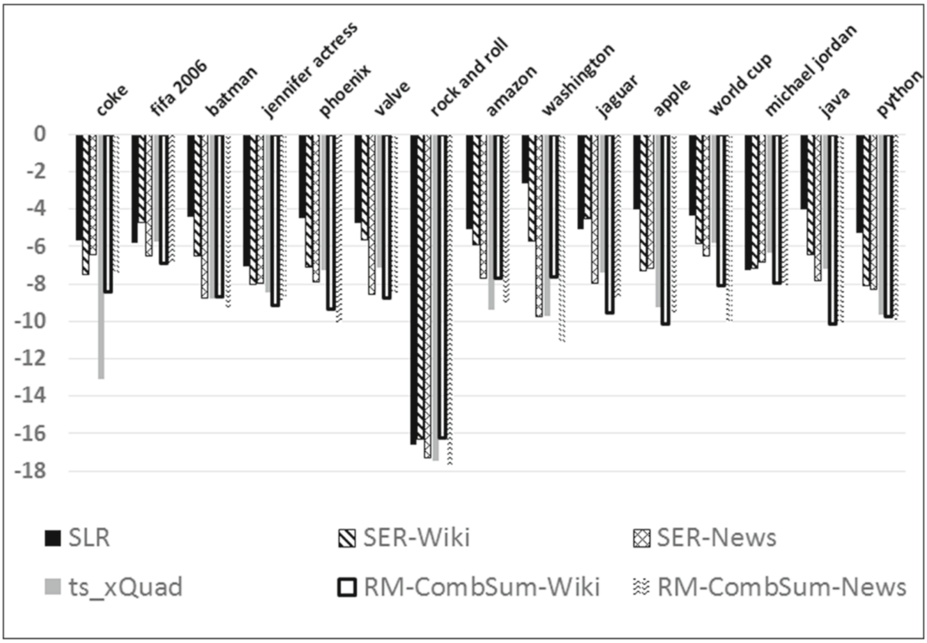


Figure 5 Jaccard Similarity based Diversity Analysis

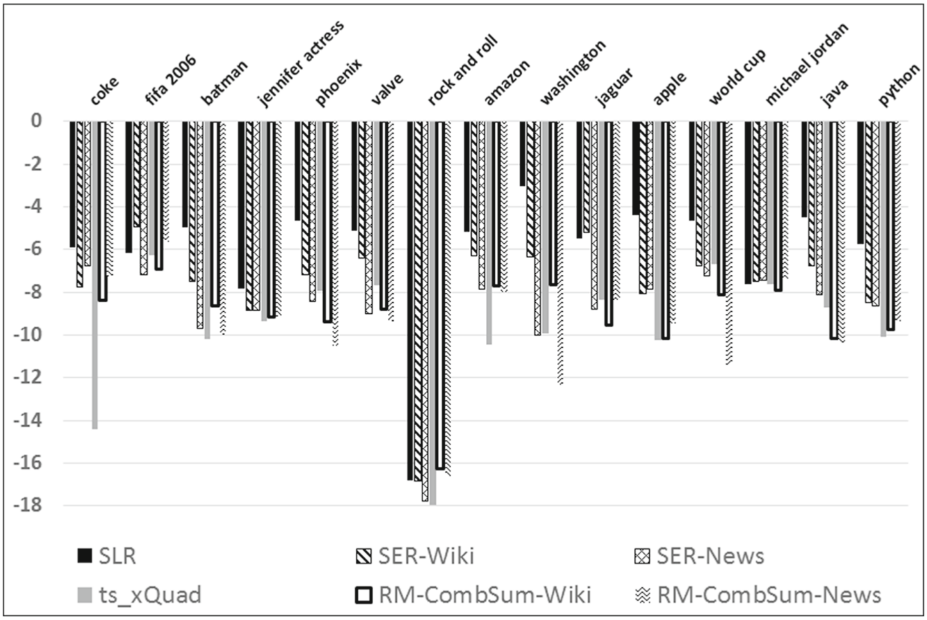


Figure 6 Dexter based Diversity Analysis

methods. It is also interesting to note that SER-News despite using word embeddings from a specialized domain (i.e., Google News) is still able to outperform ts_{xQuAD}, albeit not by much.

6.3.1 Gini index analysis

We now devise a simpler automated evaluation that does not require information about connectivity in Wikipedia or the semantic similarity estimates from Dexter. This measure is a straightforward adaptation of the Gini index,[17] a measure of statistical dispersion that has been used within data mining settings earlier (e.g., [32]). Similar to the construction of the $Q(.,.)$ measure outlined earlier, we first link each term (infact, their associated expanded queries) in $\mathbb{E}$ with entities, forming a set of entities $\mathbb{N}$ across terms in $\mathbb{E}$. An entity relevance score, as in the earlier case, is defined as $r_{\mathbb{E}}(n) = \sum_{t \in \mathbb{E}} r(t, n)$. A good quality DQE result set (i.e., a good quality $\mathbb{E}$) is expected to yield a node set $\mathbb{N}$ that (i) covers most entities that are relevant to at least one aspect of the query, and (ii) the distribution of relevance scores across entities be reasonably even (i.e., not very skewed). We now outline two Gini-index based quality measures, that differ on whether or not they use supervision in the form of a set of relevant entities to the query:

- **Unsupervised Unevenness (UU):** This measures the unevenness, using the Gini index, of relevance scores across all entities in $\mathbb{N}$.
- **Supervised Unevenness (SU):** For this, we measure the evenness of relevance scores across the entities in $\mathbb{N}^*$, a manually identified set of entities that are known to be

[17] https://en.wikipedia.org/wiki/Gini_coefficient

relevant to the query. Note that it is not necessarily the case that $\mathbb{N}^* \subseteq \mathbb{N}$ since the DQE method could potentially miss some relevant entities due to weaknesses in the method. For computing this measure, we set the relevance scores of all entities in $\mathbb{N}^* - \mathbb{N}$ to be 0.0, thus penalizing the DQE method for excluding such entities. Thus, the supervised unevenness is the Gini index measured over relevance scores of entities in $\mathbb{N}^*$. Our $\mathbb{N}^*$ is a set of 10 manually identified relevant entities to each query in our query set.

The average of UU and SU values over queries in our set are illustrated in Table 4. As the Gini index quantifies unevenness and since a fair distribution over aspects (we use distribution over entities as a proxy for it) is better, lower values are desirable. For the case where the entity relevance distribution is perfectly random (i.e., all entities have the same relevance), the Gini index would evaluate to 0.0. The trends in Table 4 indicate that SLR outperforms the others by big margins. It is interesting to note that SER-News scores better than SER-Wiki on UU while the ordering is reversed for SU; however, both of them outperform ts_{xQuAD} in both UU and SU, confirming the trends in the $Q(.,.)$ measure based analysis. We looked into the behavior of SER-News to analyze its difference across the SU and UU settings; we found that SER-News excludes certain aspects of queries that are not relevant within news contexts. For example, in the query *python*, SER-News completely avoided the programming language aspect, and thus did not bring the programming language entity within $\mathbb{N}$. Consequently, the UU Gini was evaluated only on other aspects, and thus did not penalize SER-News for such exclusions. However, for SU, since the programming language entity was among the manually identified relevant entities, it was called into operation, translating into a penalty in the SU measure. In short, the cardinality of the excluded set, i.e., $|\mathbb{N}^* - \mathbb{N}|$ was found to be significant for SER-News in some queries, explaining the difference in relative trends between the SER variants.

6.4 Parameter sensitivity analysis

We now analyze the amount of fluctuation in the results of the DQE methods when the parameter settings are varied. It is of interest to see some stability in the results when parameters are varied slightly; this would indicate that the method would be robust to changes in the character of the dataset or the external knowledge base employed. We now outline our stability analysis blueprint. First, we fetch the top-10 results of DQE from each method (SLR and SER) with the parameters set to values outlined in Section 6.1, and get their associated entities. Second, we change a particular parameter and get the entity results of the same method, and measure the overlap between the top-entities retrieved from the changed parameter settings and those from the initial parameter settings; we call this overlap as the stability factor. This is repeated for each parameter, to measure the stability of the method

Table 4 Gini Index-based analysis

Method	Unsupervised Unevenness	Supervised Unevenness
SLR	**0.465**	**0.241**
SER-Wiki	0.599	0.675
SER-News	0.553	0.703
RM-CombSum-Wiki	0.583	0.705
RM-CombSum-News	0.645	0.734
$\mathbf{ts_{xQuAD}}$	0.620	0.734

Table 5 Stability analysis

SLR			SER-Wiki		
Parameter	Range	Stability factor	Parameter	Range	Stability factor
α	0.6 – 0.7	90%	ρ	4 – 6	77%
λ	0.15 – 0.25	90%	τ	0.35 – 0.45	58%
			μ	3 – 5	65%

across each parameter in round-robin fashion. It may be noted that the measure of overlap should not be interpreted as an accuracy measure; it simply indicates the amount of deviation. In particular, a parameter variation that brings in a correct entity that was not covered by the initial parameter setting would be penalized due to divergence from the latter, thus indicating that this quality measure is not directly to accuracy measured against labelled data. We define the stability factor for a range of parameter values as the minimum among the stability values across values in the range. Table 5 lists the stability factors measured over different ranges of parameter values. As may be seen, SLR is seen to be much more stable than SER-Wiki, with the latter replacing upto two-fifths of the results with variations along τ. Overall, our methods are seen to be fairly stable against small variations in parameters.

6.5 Discussion

Our user study as well as the two automated evaluations indicate that SLR outperforms the SER variants and the baselines, with the SER variants emerging as the best alternative to SLR when a well-curated knowledge-base such as Wikipedia is not available for usage. These results indicate that our skeletal three-phase framework is effective in developing practical DQE methods. Our empirical evaluation further establishes two key properties of the proposed techniques. First, external semantic resources such as Wikipedia and word embeddings provide useful information for DQE. Second, VRRW is effective in mining accurate representatives of the various aspects related to the query. Overall, the empirical analysis establishes that our methods are effective in providing good term-level abstractions of diverse user intents.

7 Conclusions and future work

In this paper, we considered the task of leveraging external semantic resources for the Diversified Query Expansion task. We developed a three phase skeletal framework that first identifies important terms, then correlates them with external resources, and finally ranks terms to form the DQE output. Building on the framework, we developed two methods, SLR and SER, that target to exploit Wikipedia and pre-learned word embeddings for DQE respectively. Both these methods make use of VRRW, a diversity-conscious graph ranking method, for ranking terms in a diversity-conscious fashion. The SLR method, in addition to addressing diversified query expansions, is also able to directly provide a diversified entity ranking. SLR was found to be better than SER as well as other baseline methods for DQE, with SLR also improving upon the state-of-the-art in diversified entity ranking. For cases such as those where SLR is not applicable, such as specialized search domains where a well-curated and high-quality knowledge base such as Wikipedia is not available, SER is seen to be the next best method to fall back on, with the latter outperforming baseline methods such as ts_{xQuAD}. Our work establishes that external semantic resources form a very useful resource for usage in diversified query expansions, and provides effective methods for leveraging them by using diversity conscious graph ranking.

As future work, we intend to look at extending SLR and SER for specialized search tasks where the knowledge-base could have different characteristics from Wikipedia, and word embeddings are learnt over a smaller corpus, respectively. Another direction that we are currently interested is that of a graph-based visualization of DQE results and entity recommendations, for easy and effective assimilation by the user.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Bouchoucha, A., He, J., Nie, J.Y.: Diversified query expansion using conceptnet. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, ACM, pp. 1861–1864 (2013)
3. Bouchoucha, A., Liu, X., Nie, J.Y.: Integrating multiple resources for diversified query expansion. In: Advances in Information Retrieval, Springer, pp. 437–442 (2014)
4. Bouchoucha, A., Liu, X., Nie, J.Y.: Towards query level resource weighting for diversified query expansion. In: Advances in Information Retrieval, Springer, pp. 1–12 (2015)
5. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 335–336 (1998)
6. Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., Trani, S.: Dexter 2.0 - an open source tool for semantically enriching data. In: Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014., pp. 417–420 (2014)
7. Clueweb. http://lemurproject.org/clueweb09/ (2009)
8. Collins-Thompson, K.: Estimating robust query models with convex optimization. In: Advances in Neural Information Processing Systems, pp. 329–336 (2009)
9. Dalton, J., Dietz, L., Allan, J.: 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 (2014)
10. Deepak, P., Ranu, S., Banerjee, P., Mehta, S.: Entity linking for Web search queries. In: Advances in Information Retrieval, Springer, pp. 394–399 (2015)
11. Diaz, F., Mitra, B., Craswell, N.: Query expansion with locally-trained word embeddings. arXiv:160507891 (2016)
12. Dou, Z., Hu, S., Chen, K., Song, R., Wen, J.R.: Multi-dimensional search result diversification. In: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, pp. 475–484 (2011)
13. Ferragina, P., Scaiella, U.: Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM, pp. 1625–1628 (2010)
14. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI, vol. 7, pp. 1606-1611 (2007)
15. He, B., Ounis, I.: Combining fields for query expansion and adaptive query expansion. Inform Process Manag **43**(5), 1294–1307 (2007)
16. He, J., Hollink, V., de Vries, A.: Combining implicit and explicit topic representations for result diversification. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 851–860 (2012)

17. Jakarta, A.: Apache lucene-a high-performance, full-featured text search engine library (2004)
18. Krishnan, A., Padmanabhan, D., Ranu, S., Mehta, S.: Select, link and rank: Diversified query expansion and entity ranking using wikipedia. In: Web Information Systems Engineering - WISE 2016 - 17th International Conference, Shanghai, China, Proceedings, Part I, pp. 157–173, doi:10.1007/978-3-319-48740-3_11 (2016)
19. Kuzi, S., Shtok, A., Kurland, O.: Query expansion using word embeddings. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ACM, New York, NY, USA, CIKM '16, pp. 1929–1932, doi:10.1145/2983323.2983876 (2016)
20. Li, C., Wang, H., Zhang, Z., Sun, A., Ma, Z.: Topic modeling for short texts with auxiliary word embeddings. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, SIGIR '16, pp. 165–174, doi:10.1145/2911451.2911499 (2016)
21. Liu, X., Bouchoucha, A., Sordoni, A., Nie, J.Y.: Compact aspect embedding for diversified query expansions. In: Proceedings of AAAI, vol. 14, pp. 115–121 (2014)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
23. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the Web. In: Proceedings of the 7th International World Wide Web Conference, pp. 161–172 (1998)
24. Pemantle, R.: Vertex-reinforced random walk. Probab. Theory Relat. Fields **92**(1), 117–136 (1992)
25. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, http://www.aclweb.org/anthology/D14-1162 (2014)
26. Santos, R.L., Macdonald, C., Ounis, I.: Exploiting query reformulations for Web search result diversification. In: Proceedings of the 19th international conference on World wide Web, ACM, pp. 881–890 (2010a)
27. Santos, R.L., Peng, J., Macdonald, C., Ounis, I.: Explicit search result diversification through subqueries. In: Advances in information retrieval, Springer, pp. 87–99 (2010b)
28. Schuhmacher, M., Ponzetto, S.P.: Knowledge-based graph document modeling. In: Proceedings of the 7th ACM international conference on Web search and data mining, ACM, pp. 543–552 (2014)
29. Singh, A., Raghu, D., et al.: Retrieving similar discussion forum threads: a structure based approach. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 135–144 (2012)
30. Song, R., Luo, Z., Wen, J.R., Yu, Y., Hon, H.W.: Identifying ambiguous queries in Web search. In: Proceedings of the 16th international conference on World Wide Web, ACM, pp. 1169–1170 (2007)
31. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligent Analysis, Citeseer, vol. 2, pp. 2–6 (2005)
32. Telang, A., Deepak, P., Joshi, S., Deshpande, P., Rajendran, R.: Detecting localized homogeneous anomalies over spatio-temporal data. Data Min. Knowl. Discov. **28**(5-6), 1480–1502 (2014). doi:10.1007/s10618-014-0366-x
33. Van Deursen, A.J., Van Dijk, J.A.: Using the internet: Skill related problems in users' online behavior. Interact. Comput. **21**(5), 393–402 (2009)
34. Vargas, S., Santos, R.L., Macdonald, C., Ounis, I.: Selecting effective expansion terms for diversity. In: Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, pp. 69–76 (2013)
35. Whissell, J.S., Clarke, C.L.: Improving document clustering using okapi bm25 feature weighting. Inf. Retr. **14**(5), 466–487 (2011)
36. Xu, Y., Jones, G.J., Wang, B.: Query dependent pseudo-relevance feedback based on wikipedia. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 59–66 (2009)
37. Zhu, X., Goldberg, A.B., Van Gael, J., Andrzejewski, D.: Improving diversity in ranking using absorbing random walks. In: HLT-NAACL, Citeseer, pp. 97–104 (2007)