Select, Link and Rank: Diversified Query Expansion and Entity Ranking Using Wikipedia

Adit Krishnan[1](✉), Deepak Padmanabhan[2], Sayan Ranu[1], and Sameep Mehta[3]

[1] IIT Madras, Chennai, India
{adit,sayan}@cse.iitm.ac.in
[2] Queen's University Belfast, Belfast, Northern Ireland, UK
D.Padmanabhan@qub.ac.uk
[3] IBM Research, New Delhi, India
sameepmehta@in.ibm.com

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. We propose a method, Select-Link-Rank, that exploits semantic information from Wikipedia to generate diversified query expansions. SLR does collective processing of terms and Wikipedia entities in an integrated framework, simultaneously diversifying query expansions and entity recommendations. SLR starts with selecting informative terms from search results of the initial query, links them to Wikipedia entities, performs a diversity-conscious entity scoring and transfers such scoring to the term space to arrive at query expansion suggestions. Through an extensive empirical analysis and user study, we show that our method outperforms the state-of-the-art diversified query expansion and diversified entity recommendation techniques.

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 [24]. 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],

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

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

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

W. Cellary et al. (Eds.): WISE 2016, Part I, LNCS 10041, pp. 157–173, 2016.
DOI: 10.1007/978-3-319-48740-3_11

that being the dominant interpretation (aka *aspect*) in the web. *Search Result Diversification (SRD)* [5,29] 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*.

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 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 [26], query logs [17], external ontologies [2,3] or the results of the initial query [26]. The aspect-affinity of each term is modeled either explicitly [17,26] 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 address the DQE and DER problems and develop a novel method, *Select-Link-Rank (***SLR***)*. Our main contributions are:

- A novel technique, *SLR*, for diversified query expansion and entity recommendation that harvests terms from initial query results and prioritizes terms and entities using the Wikipedia graph in a diversity conscious fashion. Our method does not require query logs or supervision and thus is immune to cold start issues.
- We present an empirical evaluation including a user study that illustrates that SLR's DQE results as well as the entity ranking results are much superior than those of the respective baselines. This establishes SLR as the method of choice for DQE and DER.

2 Related Work

We will start by scanning the space of SRD methods, followed by a detailed analysis of techniques for DQE/DER.

SRD: Search Result Diversification is the task of producing a result set 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:

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

In MMR, the next document 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 [29]. Since then, there have been methods to explicitly model query aspects and diversify search results using query reformulations [20], query logs [11] and click logs [15], many of which use MMR-style diversification.

DQE/DER: 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]. Table 1 summarizes the various DQE methods in literature. Drawing inspiration from recent interest in linking text with knowledge-base entities (notably, since ESA [13]), 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 [17] 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 method, *SLR*, targets more general scenarios where query logs may not be available.

Table 1. Techniques for Diversified Query Expansion

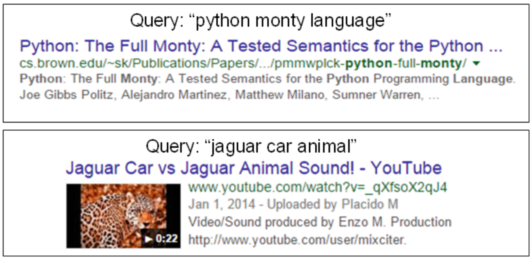
Method[a]	User Data Reqd	External Resource Reqd	Source of Exp. Terms	Remarks
BHN [2] (DER Baseline)	–	ConceptNet	Entity Names	Expansion terms from the small vocabulary of entity names
ts_{xQuAD}[26] (DQE Baseline)	Sub-topics (i.e., aspects) and sub-topic level relevance judgements	–	**Documents**	Relevance judgements are often impractical to get, in real systems
LBSN [17]	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	

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

ts_{xQuAD}[26], 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.

Wikipedia for Query Expansion: Apart from BLN, there has been previous work on using Wikipedia for Query expansion, such as [28]. This work uses Wikipedia documents, differently weighted by the structure of Wikipedia documents, in a pseudo-relevance feedback framework; it may be particularly noted that, unlike the approaches discussed so far, this work does not address the diversity factor.

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 terms[4] 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 may be a good model for search engines that work on a small corpus, 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.



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
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.

Fig. 1. Sample results from extended queries

3 Problem Formulation

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 [27] or PageRank [18]. The ideal $\mathbb{E}$ is

[4] Terms may have associated weights.

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; 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.

4 Select-Link-Rank: Our Method

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 starts 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, 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. 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 $\mathcal{Q}$, 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 [14], a popular informativeness measure

[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/.

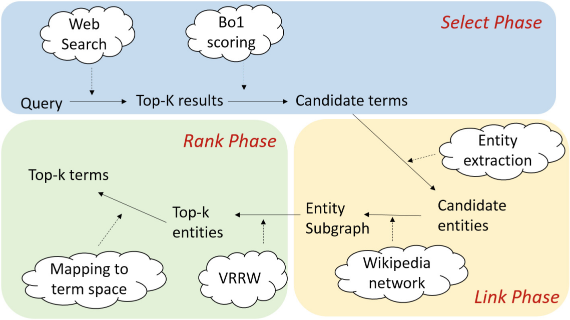


Fig. 2. Pipeline of the SLR algorithm.

that uses Bose-Einstein statistics to quantify divergence from randomness as below:

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

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 $\mathbb{D}$. To ensure all aspects of $\mathcal{Q}$ 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 [5,23].

4.2 Link: Linking to Wikipedia Entities

In this phase, we link each term in $Cand(Q, \mathbb{D})$ to one or more related Wikipedia entities. 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 $\mathcal{Q}$. We leverage entity linking methods, such as TagMe [12] and [10], which match small text fragments with entity descriptions in Wikipedia to identify top-related entities. At the end of this phase each term t in $Cand(Q, \mathbb{D})$ is associated with a set of entities, $t.E$. 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.

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.

4.3 Rank: Ranking Candidate Terms

This phase forms the crux of our method and comprises four sub-phases.

Wikipedia Subgraph Creation: In this phase, we first construct a subgraph $G(\mathcal{Q}) = \{V(\mathcal{Q}), E(\mathcal{Q})\}$ of the Wikipedia entity network $W = \{V_W, E_W\}$. In W, each Wikipedia page (entity) is a node in V_W and there is a directed edge $(e, e') \in E_W$ if an outward hyperlink from $e \in V_W$ to $e' \in V_W$ exists. $G(\mathcal{Q})$ is a subgraph of W spanning entities that are linked to terms in $Cand(Q, \mathbb{D})$ and their directly related neighbors. More specifically, $V(\mathcal{Q}) = N_1 \cup N_2$ where

$$N_1 = \{\cup_{t \in Cand(Q,\mathbb{D})} t.E\} \tag{1}$$

$$N_2 = \{e \mid \exists e' \in N_1,\ e \notin N_1,\ (e', e) \in E_W\} \tag{2}$$

The edge set $E(\mathcal{Q})$ is the set of all edges (i.e., 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. 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, subsumes inclusion of all nodes along two-hop paths between nodes in N_1; the latter heuristic has been used in knowledge graph expansion in [22]. 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.

Entity Importance Weights: In this sub-phase, we set a weight to each node (i.e., entity) in $G(\mathcal{Q})$ based on its estimated importance. We start with assigning weights to entities that are directly linked to terms in $Cand(Q, \mathbb{D})$:

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

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:

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

Thus, the weight of nodes in N_2 is set to that of their highest scored[10] inward neighbor in N_1, followed by normalization. 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:

$$wt(e) = \begin{cases} \alpha \times wt'(e) & e \in N_1 \\ (1-\alpha) \times wt'(e) & e \in N_2 \end{cases} \tag{3}$$

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.

Vertex Reinforced Random Walk: Our goal in this step 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 that does a *vertex reinforced random walk (VRRW)* [19]. 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 Fig. 3, wherein the weights are redistributed to a more mutually diverse set of nodes.

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.

$$p_T(e, e') = \lambda\ wt(e') + (1-\lambda)\frac{wt(e')N_T(e')}{D_T(e)} \tag{4}$$

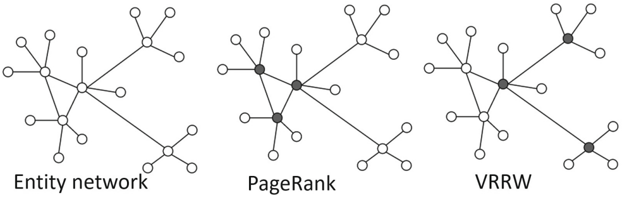


Fig. 3. The three nodes (shaded) with the highest scores in PageRank vis-a-vis VRRW.

[10] 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.

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 $S(\cdot)$ after a large T, i.e., $S(e') = \sum_{e \in V(\mathcal{Q})} p_T(e, e') S(e)$ [19]. 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*.

Why does VRRW favor representativeness? 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 Fig. 3.

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 $\mathbb{E}.E$. At each step, the next term to be added to $\mathbb{E}$ is chosen as follows:

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

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*[11], *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.

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

Algorithm 1. ***Select-Link-Rank***

```
Input: Query $\mathcal{Q}$, corpus $\mathbb{D}$
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 $r(t, e)$
Rank Phase
5. Construct $G(\mathcal{Q})$, graph of linked entities and neighbors
6. Score each entity using relatedness to linked terms
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.
```

4.4 Summary and Remarks

The various steps in SLR and their sequence of operation are outlined in the pseudocode in Algorithm 1. It may be noted that we do not make use of wikipedia disambiguation pages in SLR.

5 Experiments

Experimental Setup. We use the ClueWeb09 [7] Category B dataset comprising 50 million web pages in our experiments. In SLR, 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 [16]; specifically, all entities were indexed by text fields. For parameters, we set $K = K' = 1000$, $\alpha = 0.65$ and $\lambda = 0.25$ unless mentioned otherwise. 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} [26] 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.

Our primary evaluation is based on a user study where users are requested to choose from between our method and the baseline when shown the top-5 results from both. The user study was rolled out to an audience of up to 100 technical people (grad students and researchers) of whom around 50 % responded. All questions were optional; thus, some users only entered responses to a few of the queries. 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.

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

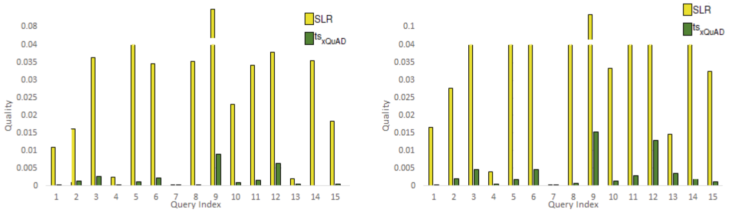
Query Information		DQE Expansions Eval.		DER Entities Eval.	
Sl#	Query	SLR	ts_{xQuAD}	SLR	BHN
1	coke	**37**	6	**40**	11
2	fifa 2006	**40**	3	**33**	18
3	batman	**32**	11	**49**	2
4	jennifer actress	**40**	3	**48**	3
5	phoenix	**39**	4	**42**	10
6	valve	**38**	5	**40**	12
7	rock and roll	**40**	3	**46**	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 %

5.1 User Study

Expansion Quality Evaluation (DQE). First, we compare the quality of SLR results against those of ts_{xQuAD} over the dataset of 15 queries. For each query, we generate the top-5 recommended expansions by both methods and request users to choose the method providing better recommendations. The number of votes gathered by each technique is shown in Table 2. The exact recommended expansions, along with all details of the user study, can be found at a web page[12]. SLR is seen to be preferred over ts_{xQuAD} across all queries.

Entity Quality Evaluation (DER). We compare the DER output from SLR against the entity ranking from BHN. We follow a similar approach as in the expansion evaluation to elicit user preferences. Table 2 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 also had results pertaining to a British comedy group, *Monty Python*; we suspect most users were

[12] https://sites.google.com/site/slrcompanion2016/.



(a) Jaccard Similarity

(b) Dexter Similarity

Fig. 4. Diversity Analysis, SLR vs ts_{xQuAD}

unaware of that aspect for python, and thus did not credit SLR for considering that.

5.2 Automated Diversity Evaluation

We further evaluate the performance of SLR 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., [4,21]), 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. 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 expansions; i.e., $r_{\mathbb{E}}(n) = \sum_{e \in \mathbb{E}} r(e, n)$. Let $S(n_i, n_j)$ denote an entity-pair semantic relatedness estimate from an external oracle; our quality measure is:

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

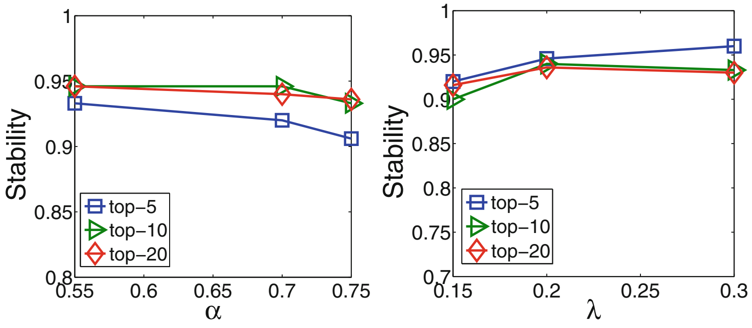
where $exp(-S(n_i, n_j))$ is a positive value inversely related to similarity between the corresponding entities. Intuitively, it is good to have highly relevant entities to be less related to ensure that entity-nodes in $\mathbb{N}$ are diverse. Thus, *higher values* of the $Q(.,.)$ metric are desirable.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

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

$$S_D(n_i, n_j) = Dexter(n_i, n_j)$$

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 4(a) and (b) show the expansion qualities based on Jaccard and Dexter respectively for the SLR and ts_{xQuAD} methods. As can be seen, regardless of the parameter values, or the quality metric used, SLR consistently outperforms ts_{xQuAD} by a significant margin. Infact, for some cases, SLR outperforms it by such a large margin that the corresponding bars in the figures been segmented for better visualization.



(a) Against α

(b) Against λ

Fig. 5. Stability of the SLR algorithm.

5.3 SLR Parameter Sensitivity Analysis

Finally, we analyze the *stability* of SLR DQE against the two parameters that it requires: the teleportation probability λ, and the weighting factor α. Stability is defined as the fraction of common recommendations in the top-20 expansions produced at two different parameter values. We consider the default setting as reference, and measure stability against of results at altered parameter values against the reference. The results in Figs. 5(a) and (b) indicate that SLR is stable across wide variations of both parameters, achieving a stability of up to 0.95. Similar trends were recorded for SLR DER.

5.4 Computational Cost Analysis

Although computational efficiency is not the focus of this work, we attempt to provide a brief analysis of the computational costs of our algorithm.

- The **Select** phase uses the Indri Search Engine to run the queries, which combines language modeling and inference network approaches to perform the search. Interested readers may refer [25] for performance numbers. Selection of K' terms from K retrieved documents can be performed using a heap, at a cost of $K.L_{avg} + W_u.log(K')$, where L_{avg} is the mean count of non stop-words per document and W_u is the total number of unique words.

- In the **Link** phase, each of the K' 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. 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).
- Under the **Rank** phase, let us consider a subgraph of size $|S|$ nodes, on which DivRank is executed. With the matrix implementation of DivRank, the total computational cost is $\propto |S|^2$ per iteration. In practice, we found all our subgraphs to reasonably converge in less than 15 iterations, leading to very fast computations in the order of a few seconds.

5.5 Discussion

Our user study on both expansions and entities indicate that SLR results outperform other methods. SLR is also seen to perform better on automated diversity evaluation measures. These results establish two key properties of the proposed technique. First, the Wikipedia entity network is a meaningful resource to understand the various aspects of a query. Second, VRRW is effective in mining accurate representatives of the various aspects related to the query. Overall, the empirical analysis establishes that entities may be leveraged towards providing good term-level abstractions of diverse user intents.

6 Conclusions and Future Work

In this paper, we considered the problem of Diversified Query Expansions and developed a method that leverages semantic information networks such as Wikipedia towards providing diverse and relevant query expansions. Our method, SLR, exploits recent technical advancements across fields such as entity analysis, NLP and graph traversals using a simple 3-phase select-link-rank framework. The SLR query expansion and entity recommendations were seen to outperform respective baselines by large margins, on a user study as well as on an automated diversity evaluation. These establish SLR as the method of choice for DQE and diversified entity recommendations. As future work, we intend to look at extending SLR to exploit structured domain-specific knowledge sources to enhance usability for specialized scenarios such as intranet search. We are currently exploring integrated graph-based visualization of DQE results and entity recommendations.

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 and Knowledge Management, pp. 1861–1864. ACM (2013)
3. Bouchoucha, A., Liu, X., Nie, J.-Y.: Integrating multiple resources for diversified query expansion. In: Rijke, M., Kenter, T., Vries, A.P., Zhai, C.X., Jong, F., Radinsky, K., Hofmann, K. (eds.) ECIR 2014. LNCS, vol. 8416, pp. 437–442. Springer, Heidelberg (2014). doi:10.1007/978-3-319-06028-6_38
4. Bouchoucha, A., Liu, X., Nie, J.-Y.: Towards query level resource weighting for diversified query expansion. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 1–12. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16354-3_1
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, pp. 335–336. ACM (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 and 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: (2009). http://lemurproject.org/clueweb09/
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 and Development in Information Retrieval, pp. 365–374. ACM (2014)
10. Deepak, P., Ranu, S., Banerjee, P., Mehta, S.: Entity linking for web search queries. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 394–399. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16354-3_43
11. 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, pp. 475–484. ACM (2011)
12. 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, pp. 1625–1628. ACM (2010)
13. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJCAI **7**, 1606–1611 (2007)
14. He, B., Ounis, I.: Combining fields for query expansion and adaptive query expansion. Inf. Process. Manage. **43**(5), 1294–1307 (2007)
15. 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, pp. 851–860. ACM (2012)
16. Jakarta, A.: Apache lucene-a high-performance, full-featured text search engine library (2004)
17. Liu, X., Bouchoucha, A., Sordoni, A., Nie, J.Y.: Compact aspect embedding for diversified query expansions. Proc. AAAI **14**, 115–121 (2014)
18. 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)

19. Pemantle, R.: Vertex-reinforced random walk. Probab. Theor. Relat. Fields **92**(1), 117–136 (1992)
20. 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, pp. 881–890. ACM (2010)
21. Santos, R.L.T., Peng, J., Macdonald, C., Ounis, I.: Explicit search result diversification through sub-queries. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., Rijsbergen, K. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 87–99. Springer, Heidelberg (2010). doi:10.1007/978-3-642-12275-0_11
22. Schuhmacher, M., Ponzetto, S.P.: Knowledge-based graph document modeling. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 543–552. ACM (2014)
23. 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, pp. 135–144. ACM (2012)
24. 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, pp. 1169–1170. ACM (2007)
25. 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. vol. 2, pp. 2–6. Citeseer (2005)
26. 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)
27. Whissell, J.S., Clarke, C.L.: Improving document clustering using okapi bm25 feature weighting. Inf. Retr. **14**(5), 466–487 (2011)
28. 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, pp. 59–66. ACM (2009)
29. Zhu, X., Goldberg, A.B., Van Gael, J., Andrzejewski, D.: Improving diversity in ranking using absorbing random walks. In: HLT-NAACL, pp. 97–104. Citeseer (2007)