Stochastic Re-Ranking of Biomedical Search Results based on Extracted Entities

Pavlos Fafalios* and Yannis Tzitzikas

Institute of Computer Science, FORTH-ICS, Greece, and
Computer Science Department, University of Crete, Greece
{fafalios, tzitzik}@ics.forth.gr

Abstract

Health-related information is nowadays accessible from many sources and is one of the most searched-for topics on the internet. However, existing search systems often fail to provide users with a good list of medical search results, especially for classic (keyword-based) queries. In this paper we elaborate on whether and how we can exploit biomedicine-related entities from the emerging Web of Data for improving (through re-ranking) the results returned by a search system. The aim is to promote relevant but low-ranked hits containing entities that are important to the current search context. We introduce an approach that is based on entity extraction applied on the retrieved documents, yielding a graph of documents along with entities which in turn is analyzed probabilistically using a Random Walk-based method. The proposed approach is independent of the submitted query and the underlying retrieval models, and thus can be applied over any ranked list of medical search results. Evaluation results using the dataset of TREC Clinical Decision Support track demonstrate that the proposed approach can significantly improve the results returned by classic and widely applicable retrieval models. The results also enabled us to identify cases where the proposed re-ranking method fails to improve the ranking.

Introduction

The increasing availability of biomedical information in electronic form has made such data accessible to a wide variety of users including clinicians, practitioners, researchers, as well as patients and their families. Health-related content is nowadays one of the most searched-for topics on the Internet [Goeuriot et al., 2014]. However, effective access and retrieval of such information remains particularly challenging [Roberts et al., 2015]. Existing search systems often fail to retrieve and provide to the user a good list of search results, especially for classic (keyword-based) search queries. This can happen because, for example, the query submitted by a typical user is not good or expressive enough. In such cases, even a very effective search system may return search hits of low quality.

At the same time, the Web is not anymore a single "world" of unstructured documents. It now comprises two webs: the Web of Documents (containing mainly Web pages) and the Web of Data (structured data in the RDF format), also called Linked Data [Heath and Bizer, 2011]. There are several ways to establish "bridges" between these two worlds. In this paper we investigate an entity-based approach, i.e., an approach where *named-entities* (like names of persons, locations, drugs, diseases, chemical substances, etc.) are used as the "glue" for automatically connecting documents with data and knowledge. Such an entity-based integration can be useful in many different contexts. For instance, *entity-based faceted search* [Fafalios and Tzitzikas, 2014a] allows a user to quickly locate results containing information about one or more particular entities. As another example, *Google Knowledge Graph* [Singhal, 2012] allows users to quickly get fresh information related to their search context without disengaging from their initial search task (e.g., photos and main characteristics of an entity).

In this paper, we investigate whether and how this integration can be exploited for *re-ranking* a list of biomedicine-related search results. The objective is to improve the search results by promoting low-ranked but relevant hits containing biomedical entities (like *diseases* and *drugs*) that are important

*Since June 2016, Pavlos Fafalios has moved to L3S Research Center (University of Hannover, Germany) as a postdoctoral researcher.

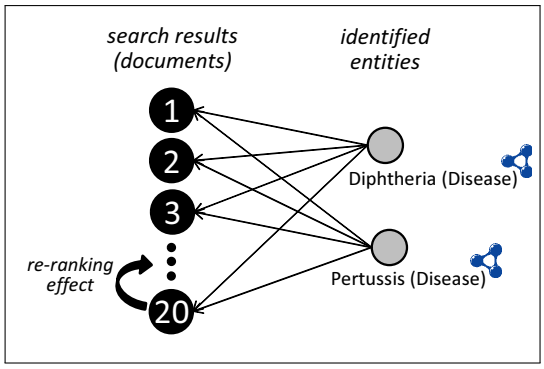


Figure 1: The effect of the proposed re-ranking method.

for the current search context. The proposed approach is independent of the submitted query and the underlying Information Retrieval (IR) model, and thus can be applied over any ranked list of biomedical search results. The idea is to construct a graph of documents (actually of search results) and extracted entities, and then to analyze it probabilistically. For analyzing the graph and scoring its nodes, we follow a biased Random Walk-based method (PageRank-like), while for reaching to a widely applicable approach we investigate a scenario where this analysis is performed at query time with no human effort. The objective is to compute the probabilities the random walker to be in each document-node. These probabilities actually define a new ranking for the search results. A low-ranked document containing some highly-scored entities will receive a high score and thus it will be high in the new ranked list of results.

Figure 1 depicts an indicative example. We notice that two entities of type disease were identified in the top-3 returned documents and thereby can be considered important for the current search context. However, these entities were also detected in the 20th document. After applying the proposed re-ranking method, the 20th document will be ranked higher in the new list of results because it contains two "important" (highly-scored) entities. Thus, such a functionality can help physicians to locate faster bigger number of relevant documents, since low-ranked but relevant results are promoted in higher positions.

Experimental results over the dataset of the TREC Clinical Decision Support track [Roberts et al., 2015] demonstrate that the proposed re-ranking approach can notably improve the list of results returned by two classic and widely applicable IR models. Paired t-tests with α-level 5% indicate that the improvement of the re-ranked list is statistically significant in almost all cases. However, additional semantic information about the entities (like properties and related entities) can mislead the random walker and affect negatively the results. The same can happen when the initial answer is very good (containing a big number of relevant hits) or very bad. Nevertheless, the results showed that even if the number of relevant hits is very high, re-ranking can further increase recall.

Related Work

The last years, the biomedical and healthcare domain has attracted the interest of the IR research community. This is evident from the emergence of several venues dedicated to medicine-related IR, like the Medical IR Workshops (2014, 2016) [Goeuriot et al., 2014], the TREC Clinical Decision Support track (2014-2016) [Roberts et al., 2015], and the CLEF eHealth IR tasks (2013-2016) [Palotti et al., 2015]. These venues have led to a large body of works on topics related to medical IR with the aim to improve access to medical information.

Below we review the main works on the general problem of improving automatically the results returned by a search system. The works can be classified in three main categories: *automatic query expansion*, *pseudo-relevance feedback*, and *re-ranking*. Some of the discussed works are related to the biomedicine domain. At the end we describe a highly-related work which applies a common entity-based probabilistic analysis in the context of exploratory search.

Automatic Query Expansion

Automatic query expansion is the process of reformulating a query to improve retrieval performance without any user interaction. Specifically, the original query submitted by a user is automatically expanded with other words that best capture the actual user intent, or that simply produce a more useful query,

i.e., a query that is more likely to retrieve relevant documents. The computational steps involved mainly include data acquisition and preprocessing, candidate feature generation and ranking, feature selection, and query reformulation.

[Carpineto and Romano, 2012] survey approaches of automatic query expansion. The authors classify the approaches into five main groups according to the conceptual paradigm used for finding the expansion features: linguistic methods (e.g., using syntactic analysis [Sun et al., 2006]), corpus-specific statistical approaches (e.g., making use of latent semantic indexing [Park and Ramamohanarao, 2007]), query-specific statistical approaches (e.g., using document summaries of top retrieved documents [Chang et al., 2006]), methods that exploit search log analysis (e.g., by extracting terms from clicked results [Riezler et al., 2007]), and methods using Web data (e.g., using categories of Wikipedia articles [Xu et al., 2009]). In the biomedicine domain, [Babashzadeh et al., 2013] use semantic information to improve the performance of clinical IR systems by representing queries in an expressive context. The authors model and develop a query domain ontology which represents concepts closely related to the query. The query context is then exploited in query expansion and patients records re-ranking for improving clinical retrieval performance.

Pseudo-Relevance Feedback

Relevance feedback analyzes the results that are initially returned from a given query and uses information (provided by the user) about whether or not those results are relevant to perform a new query. Pseudo-relevance feedback automates the manual part of relevance feedback so that the user gets improved retrieval performance without any interaction.

Pseudo-relevance feedback has been widely used in IR and has been implemented in different retrieval models. [Lee et al., 2008] propose a cluster-based re-sampling method to select better relevant documents based on the relevance model. The idea is to use document clusters to find dominant documents from the initial retrieval set. [Tao and Zhai, 2006] present a method based on statistical language models. [Cao et al., 2008] propose the integration of a term classification process to predict the usefulness of expansion terms, while [Xu et al., 2009] exploit Wikipedia entity pages for query-dependent pseudo-relevance feedback. Finally, a recent work related to medical IR incorporates the structure of external collections for estimating individual components in the final feedback model [Oh and Jung, 2015].

Re-ranking

Re-ranking aims to improve the original list of results by reordering the returned hits. [Chidlovskii et al., 2000] present a collaborative re-ranking system architecture for integrating user and community profiling to the information search process. [Kanhabua and Nørvåg, 2010] propose a number of methods to determine the time of queries using temporal language models and shows how to increase the retrieval effectiveness by using the determined time of queries to re-rank the search results. [Zhuang and Cucerzan, 2006] introduce a method, called Q-Rank, to effectively refine the ranking of search results for any given query by constructing the query context from search query logs. In the biomedicine domain, there are two works which focus on the *diversity* aspect. [Yin et al., 2010] propose a cost-based re-ranking method to promote diversity for biomedical IR. The proposed method focuses on finding passages that cover many different aspects of a query topic. Aspects covered by the retrieved passages are detected and explicitly presented by Wikipedia concepts. The detected aspects are ranked in decreasing order of the probability that an aspect is generated by the query and the retrieved passages are finally re-ranked using a cost-based re-ranking method which considers the number of new aspects covered by the passage as well as their query-relevance. [Li et al., 2015] exploit a domain ontology to extract the semantic information implied in a user query and to model query aspects. Then, based on the modeled query aspects, a diversification strategy is proposed to perform document ranking which considers both the aspect importance and the aspect similarity.

The approach that we propose also falls in this category. However, our work does not require access to user profiles or query logs. Note that such information often is not available, especially in the "privacy preserving" context of a medical IR system. The proposed method relies only on information extracted dynamically from the retrieved results and is independent from the underlying IR model and the submitted query (i.e., independent from the way the list of results was produced). Furthermore, the two biomedicine-related works focus on a different task (diversity of search results), while the re-ranking method that we propose is applicable to classic "ad-hoc" search.

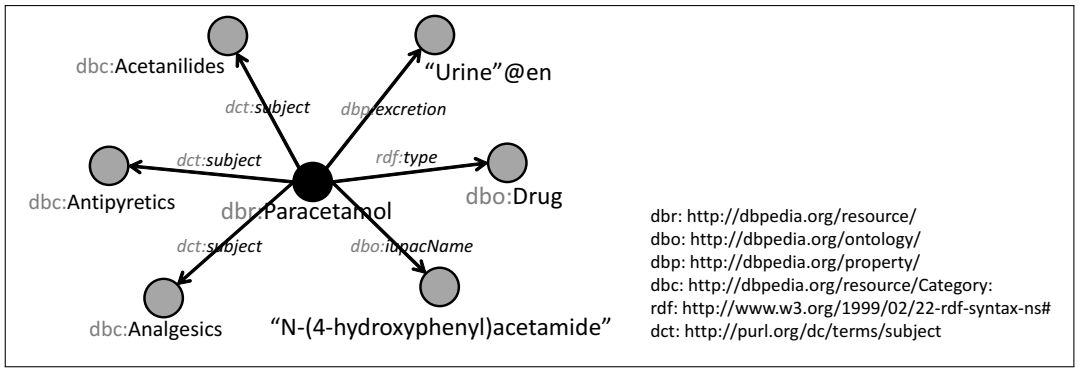


Figure 2: Example of a small RDF graph describing information about the DBpedia entity "Paracetamol".

Stochastic Analysis of Search Results based on Extracted Entities

[Fafalios and Tzitzikas, 2014b, Fafalios et al., 2014b] introduced a probabilistic post-analysis process for exploratory searching in which the search results are connected with data and knowledge at query time with no human effort. For identifying the semantic information (entities and properties) that better characterizes the search results, a Random Walk-based ranking model is introduced for the problem at hand which is exploited for producing and showing to the user *top-K semantic graphs*. A top-K semantic graph can complement the query answer with useful information regarding the connectivity of the identified entities, allowing the user to instantly inspect semantic information that may exist in different places and that may be laborious and time consuming to locate.

In this work, we continue this line of research and we investigate whether and how such "overview" information (entities detected in the results and semantic information associated to these entities) can be exploited for producing a better ranking for the retrieved results.

Context

In this section, we first define the basic concepts and then we describe the steps of the considered search process.

Entities of Interest and Semantic Knowledge Bases

Entities of Interest (EoI) are names of entities belonging to pre-defined categories/classes that are important in the application context. In our biomedical search scenario, the EoI may include names of drugs, diseases, proteins, and chemical substances. Correspondingly, the EoI of a marine-related search system may be names of fish species, water areas, and countries. The current trend is to publish information about entities in semantic knowledge bases as Linked Open Data (LOD) [Heath and Bizer, 2011].

A Semantic Knowledge Base (SKB) is an RDF dataset accessible as LOD or through a SPARQL Protocol service (called SPARQL endpoint). Examples of such publicly available SKBs are DBpedia and DrugBank. In general, such SKBs contain plenty of information for several types of named-entities.

Now we formalize the structured knowledge available in a SKB. Consider an infinite set U of RDF URI references, an infinite set B of blank nodes and an infinite set L of literals. A triple $(s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$ is called an *RDF triple* (s is called the *subject*, p the *predicate* and o the *object*). A SKB, or equivalently an *RDF graph* G, is a set of RDF triples. For an RDF Graph G_i we shall use U_i, B_i, L_i to denote the URIs, blank nodes and literals, respectively, that appear in the triples of G_i. Figure 2 depicts an example of a small RDF graph containing 7 nodes and describing information about the DBpedia entity "Paracetamol" (through 6 RDF triples).

The Search Process

Figure 3 depicts the steps of the search process that we consider. First, the user submits a query to a search system and the top-L results are retrieved. Then, an entity extraction and linking system is exploited for detecting entities in the retrieved results and linking them to web resources (e.g., DBpedia URIs). More semantic information about the extracted entities can be optionally retrieved by accessing one or more SKBs. A graph containing as nodes i) the top-L *documents* retrieved by the underlying

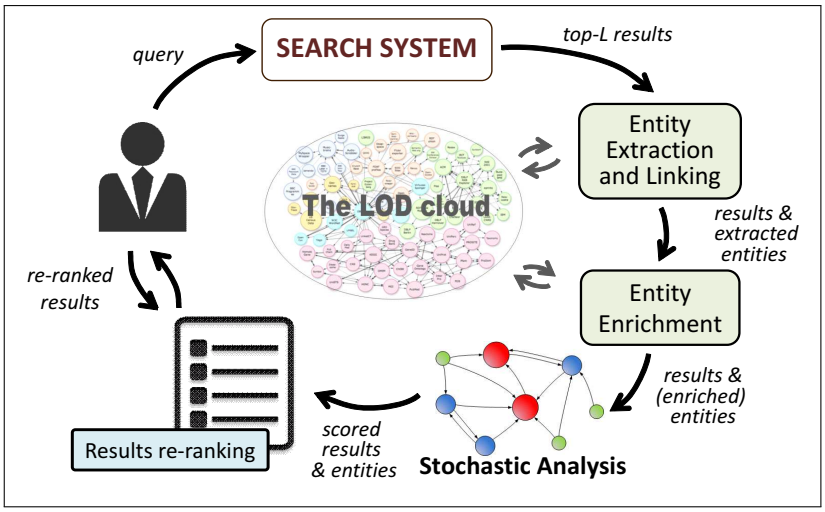


Figure 3: The considered search process.

corpus, ii) *entities* identified in the top-L documents, and optionally iii) *semantic information* related to the identified entities, is constructed and analyzed probabilistically using a Random Walk-based method. The probabilistic analysis assigns a final PageRank-like score to each graph node. Finally, the documents are re-ranked according to their final scores and the new ranked list of results is returned to the user.

Stochastic Analysis

In this section, we first define the main notions and notations and the re-ranking problem, then we introduce the notion of *"entity importance"*, and finally we detail the probabilistic analysis process.

Notions and Notations

Assuming that we are in the context of a submitted query q, we define the following notions and notations:

Documents (search results)

- L: number of top documents (hits) to retrieve from the underlying search system for the query q.
- A: the ranked list of the top-L documents of the answer of q.
- $score(a)$: the score (value in the range $[0, 1]$) of a document $a \in A$ as returned by the underlying search system for the query q.
- $rank(a)$: the position of a document $a \in A$ in the answer for the query q (i.e., the first hit has rank equal to 1, the second 2, etc.).

Document parts

- P: the set of different "parts" that constitute a document, e.g., $P = \{title, abstract, body\}$.
- a_p: the part of document $a \in A$ of type $p \in P$ (e.g., its *abstract*).
- $w(p)$: the weight expressing the importance of a part $p \in P$, where $\sum_{p' \in P} w(p') = 1$. For example: $w(title) = 0.5$, $w(abstract) = 0.3$, $w(body) = 0.2$.

Mined entities

- $ent(a_p) \subseteq EoI$: the set of entities extracted from the part p of a document $a \in A$.
- $ent(a) = \cup_{p \in P} ent(a_p)$: the set of all entities extracted from a document a (from all its parts).
- $E = \cup_{a \in A} ent(a)$: the set of all entities extracted from the list of documents A.
- $docs(e) = \{\ a \in A \mid e \in ent(a)\}$: the elements of A in which e has been detected (inverse of $ent(a)$).
- $ef(e, a_p)$: the frequency (number of occurrences) of the entity e in the part p of the document a.

Problem Definition

Given a ranked list of search results A, a set of entities of interest EoI, and a reference RDF graph G (i.e., a SKB), the *stochastic re-ranking* task aims at deriving a new (hopefully improved) ranking for the results in A by taking into consideration the EoI extracted from these results and their associations.

Entity Importance

We now define the notion of *"entity importance"*. As regards a single document, we consider that the more frequent entities are the more important. The term frequency (in our case *entity frequency*) is a classic numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. We also take into account the different parts of a document and promote the entities identified in the important parts (i.e., in those with the highest weight). For example, regarding a scientific article, we may consider that the entities detected in the title of the article are more important than those detected in the main body. Precisely, the importance of an entity e within a document a is defined as:

$$imp(e,a) = \sum_{p \in P} \left(\frac{ef(e,a_p)}{\max\limits_{e' \in ent(a_p)} ef(e',a_p)} \cdot w(p)\right) \quad (1)$$

Now, as regards the importance of an entity in the whole set of top-L retrieved documents, we consider that the top-scored results probably contain more useful entities than the low-scored results (since they are considered better results for the user query). Thereby, the importance of an entity e in the set of retrieved documents is defined as:

$$HitScore(e) = \sum_{a \in docs(e)} (imp(e,a) \cdot score(a)) \quad (2)$$

In case the document scores are not given by the underlying search system, we can use the following more generic formula which takes into account only the ranking of the retrieved documents.

$$HitScore(e) = \sum_{a \in docs(e)} \left(imp(e,a) \cdot \left(1 - \frac{rank(a)}{|A|+1}\right)\right) \quad (3)$$

The advantage of this formula is that it is applicable also at a meta (uncooperative) search level where the document scores are usually not provided by the search system.

Probabilistic Analysis

The idea is to construct *dynamically* a graph of documents and identified entities and then to analyze it probabilistically for identifying the important document and entity nodes. For analyzing the graph and scoring its nodes, we prefer to follow a Random Walk-based (PageRank-inspired [Page et al., 1999]) method because the underlying theoretical framework is solid (random walks and stochastic processes) and it can be customized (biased) according to the needs of different types of applications. Below we first present an exploratory searching scenario (from the user side) which allows to better motivate the Random Walk-based approach that we propose, and then we detail the probabilistic analysis.

Modeling a Random Walker

We model the exploratory search process as a *random walker* of the graph defined by the documents, the mined entities and their connections. Specifically, whenever the walker is at a document d:

(a) With probability p_1 he jumps to another document. The higher the relevance score/rank of a document is, the higher is the probability to jump to that document.

(b) With probability $1-p_1$ he moves to a node corresponding to an entity mined from d. The higher the entity importance score is (i.e., $HitScore$), the higher is the probability to move to that entity.

When at an entity e:

(c) With probability p_2 he jumps to a document (based on the document scores/ranks as in (a)).

(d) With probability $1-p_2$ he follows an edge from e, specifically:

  (e) With probability p_3 he moves to a document that contains e (based on the document scores/ranks as in (a) and (c)).

  (f) With probability $1-p_3$ he moves to a connected entity/property (equiprobably).

Figure 4 depicts the Markov chain of the corresponding stochastic process. This process actually models the behavior of a user in a Faceted Search-like environment: the user submits a query and the system returns a list of results as well as entities extracted from these results (e.g., in a left sidebar). The user can now open a result, or click on an entity and only display the results that contain the selected entity. In the latter case, the user can now either i) open one or more of the displayed results, or ii) click on some other entities and update the displayed list of results correspondingly, or iii) clear her/his selection

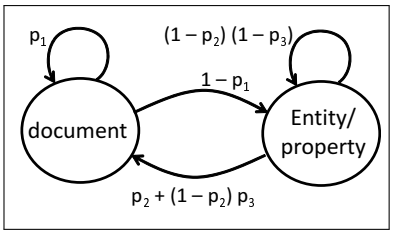


Figure 4: The Markov chain of the stochastic process.

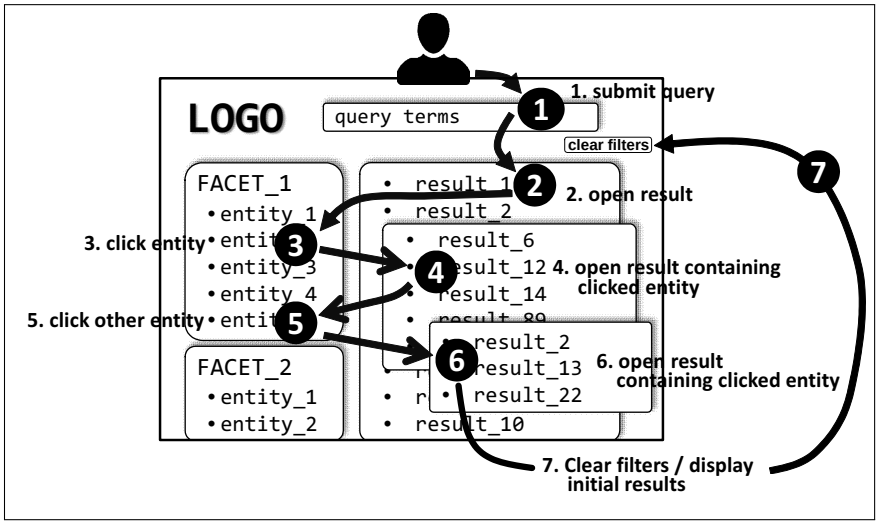


Figure 5: Example of possible user actions in a faceted search system.

(reset) and look again into the results. Figure 5 depicts the steps of such a searching scenario. Note that using fundamental PageRank in our problem (i.e., without considering the ranking of documents and the importance of entities) is impracticable since it will just favor documents containing many entities.

Creating the Graph of States and Transitions

We first define the semantic graph of documents and entities, and then we detail the construction of the graph of states and transitions that corresponds to the aforementioned modeling.

The semantic graph of documents and entities. We consider both the documents and the entities as vertices in $\mathcal{X}$, while for drawing the edges we take into account the documents in which an entity has been detected. Specifically, we draw an edge starting from an entity e and ending to a document a, if $e \in ent(a)$ (i.e., e has been extracted from a). Now, by exploiting a SKB (i.e., an RDF graph G_i), we can fetch interesting (for the search context) triples that describe information about the detected entities, like properties and related entities (recall the *entity enrichment* step in the considered search process). Let $u_e \in U_i$ be the URI of the entity e in the RDF graph G_i, and $T(u_e) \subseteq G_i$ be triples that describe information about u_e (i.e., u_e is the subject in the triple). For each entity $e \in E$ and each triple $(u_e, p, o) \in T(u_e)$, we add to $\mathcal{X}$ the vertex o and the edge $e \rightarrow o$. Furthermore, for two entities $e1, e2 \in E$, if $(u_{e1}, p, u_{e2}) \in T(u_{e1})$ or $(u_{e2}, p, u_{e1}) \in T(u_{e2})$ then we draw the corresponding edge p that connects the two entities. Figure 6 depicts an example of a semantic graph of documents and entities. The black nodes correspond to documents (set A), the gray to entities detected in the documents (set E), while the white to related properties/entities retrieved from a SKB (let us call this set R).

The State Transition Graph (STG). Now we describe how from $\mathcal{X}$ we define a STG $\mathcal{G} = (\mathcal{E}, \mathcal{P})$. For each node n in $\mathcal{X}$, we create a node in $\mathcal{G}$. For each directed edge $(n \rightarrow n')$ in $\mathcal{X}$ we create *two* directed edges in $\mathcal{G}$; one of the same direction $(n \rightarrow n')$ and one of the opposite direction $(n' \rightarrow n)$. We do that because, in our context, we consider that if a property connects two nodes in $\mathcal{X}$, then these nodes are *semantically biconnected*. For example, in the case of a document a and an entity e we can either say that $(e$, *"detectedIn"*, $a)$ or that $(a$, *"contains"*, $e)$, i.e., the difference lies in how we name the property.

Weighting the edges. In case the random walker lies in a document-node a, we consider the following formula for specifying the weights of the edges from a to entity-nodes $e \in ent(a)$:

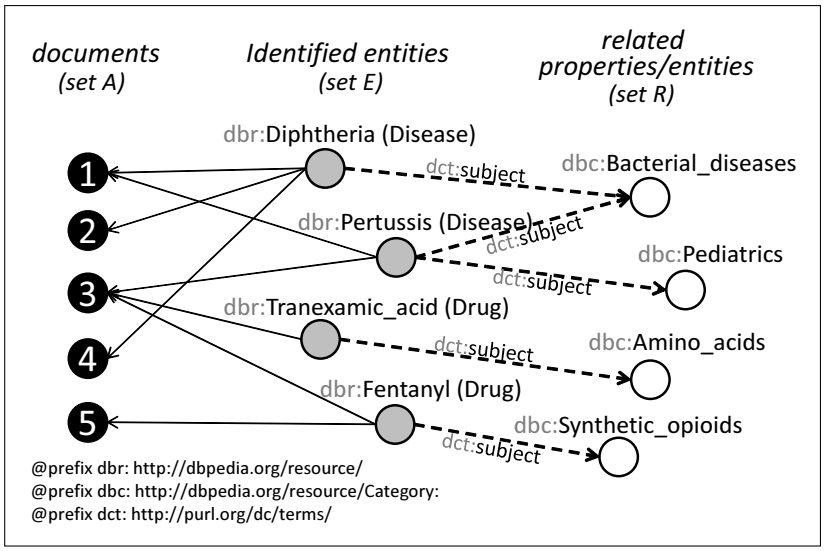


Figure 6: A medicine-related example of a semantic graph of documents and entities.

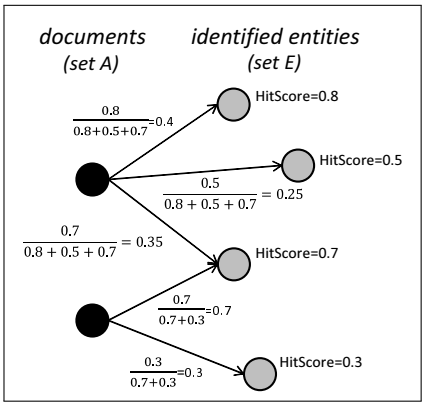


Figure 7: Biasing the link selection from a document-node.

$$weight(a \rightarrow e) = \frac{HitScore(e)}{\sum\limits_{e' \in ent(a)} HitScore(e')} \tag{4}$$

We notice that the transition probabilities are affected by the "importance" of the detected entities. Specifically, the higher the score of an entity is, the higher is the probability to move to that entity. Figure 7 depicts an example of a small STG of documents and entities showing also the edge weights as they are derived from the above formula. For simplicity and ease of comprehension, the graph includes only the edges from documents to entities.

Similarly, in case the random walker lies in an entity-node e, we consider the following formula for specifying the weights of the edges from e to document-nodes $a \in docs(e)$:

$$weight(e \rightarrow a) = \frac{score(a)}{\sum\limits_{a' \in docs(e)} score(a')} \tag{5}$$

Now the transition probabilities are affected by the similarity scores given to the documents by the underlying search system.

An entity-node may also be connected with other detected entities or with related properties/entities (result of *entity enrichment* process). In this case the weights of the edges are defined equiprobably as follows:

$$weight(e \rightarrow e') = \frac{1}{|edges_{out}(e)|} \tag{6}$$

where $edges_{out}(e)$ is the directed outgoing edges from the entity-node e to nodes that do not correspond

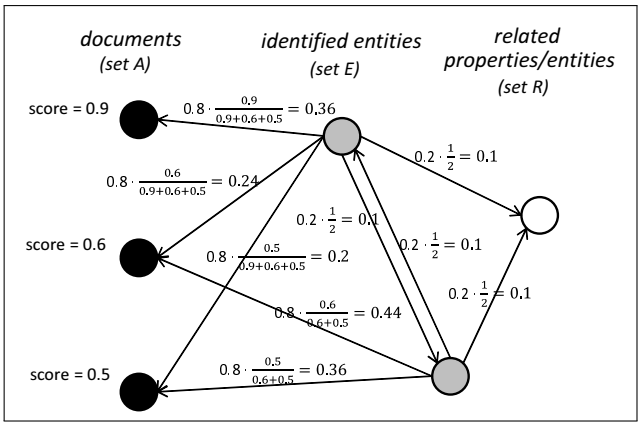


Figure 8: Biasing the link selection from an entity-node.

to documents. However, the weights of the outgoing edges of a single node must represent transition probabilities, i.e., they must sum to 1. Thus, the weight from an entity-node e to a connected node n can be generally defined as:

$$weight(e \to n) = \begin{cases} p_3 \cdot \frac{score(n)}{\sum_{a' \in A} score(a')} & n \in A \\ (1 - p_3) \cdot \frac{1}{|out(n)|} & n \notin A \end{cases} \tag{7}$$

where p_3 is the probability to select a document-node. In our context, we consider that when the walker lies in an entity-node, it is more probable to move to a document-node than to a related entity/property node (since users' main target is to locate one or more documents that satisfy their information need). Thereby, we can define $p_3 > 0.5$, e.g., $p_3 = 0.8$. Figure 8 depicts an example of a small STG showing also the edge weights as they are derived from Formula 7 with $p_3 = 0.8$ (for simplicity, the graph includes only the outgoing edges of the gray entity-nodes). Finally, when the walker lies in a related property/entity node, he can move to the connected entity-nodes equiprobably.

To sum up, the weight of the edge from a node n' to a connected node n is defined as:

$$weight(n' \to n) = \begin{cases} \frac{HitScore(n)}{\sum_{e' \in ent(n')} HitScore(e')} & n' \in A,\ n \in E \\ p_3 \cdot \frac{score(n)}{\sum_{a' \in A} score(a')} & n' \in E,\ n \in A \\ (1 - p_3) \cdot \frac{1}{|out(n')|} & n' \in E,\ n \notin A \\ \frac{1}{|out(n')|} & n' \in R,\ n \in E \end{cases} \tag{8}$$

Analyzing the STG

The objective is to find the probability the random walker to be in a specific document-node. For a node n, let $in(n)$ be the set of nodes that point to n. The PageRank-like value $r(n)$ is defined as:

$$r(n) = d \cdot Jump(n) + (1 - d) \cdot \sum_{n' \in in(n)} (weight(n' \to n) \cdot r(n')) \tag{9}$$

where d is the probability (decay factor) the walker to perform a random jump and $Jump(n)$ expresses the probability the walker to jump to the node n, and $weight(n' \to n)$ (as defined in Formula 8) is the probability the walker to visit n when being in a node n' connected to n. The values can be computed iteratively and iterations should be run to convergence. Algorithm 1 describes the corresponding PageRank-like algorithm. $\mathbf{T}$ is the matrix of the transition probabilities (i.e., the weights $weight(n' \to n)$ for each pair of nodes n and n'), $\mathbf{J}$ is the matrix of the random jumps (i.e., the probabilities $Jump(n)$ for

each node n), $\mathbf{I}$ is the matrix of the initial scores of all nodes, d is the decay factor, and N is the number of Pagerank iterations.

Algorithm 1 A PageRank-like algorithm for analyzing a state transition graph.

Require: $\mathbf{T}$ (transition matrix), $\mathbf{J}$ (random jumps matrix), $\mathbf{I}$ (matrix of initial values), d (decay factor), N (number of iterations)
Ensure: $\mathbf{r}$ (PageRank scores)
1: $\mathbf{r} = \mathbf{I}$ //initialize the score of all entities
2: **for** $i = 1$ **to** N **do**
3: $\quad \mathbf{r} = d \cdot \mathbf{J} + (1 - d) \cdot \mathbf{T} \cdot \mathbf{r}$
4: **end for**
5: **return** $\mathbf{r}$

Random jumps. As we have seen in our exploratory search scenario, we allow the random jumps only to nodes corresponding to documents. This means that the decay factor d in Formula 9 actually corresponds to the probabilities p_1 and p_2 of our "exploratory search" modeling, i.e., $p_1 = p_2 = d$. In addition, we adjust the jump probabilities according to the document scores (instead of assuming a uniform distribution). Specifically, for a node n we consider the following formula for the random jumps:

$$Jump(n) = \begin{cases} 0 & n \notin A \\ \frac{score(n)}{\sum_{a' \in A} score(a')} & n \in A \end{cases} \quad (10)$$

which means that the probability the random walker to jump to a document is higher if the document has received a high similarity score from the underlying search system.

In case the similarity scores are not provided by the search system, the above formulas can be easily adjusted to use the ranking scores, i.e., $rank(a)$, instead of similarity scores.

Tuning. To run the above Random Walk-based algorithm, it remains to tune some of its parameters:

Probabilities. At first we must specify a value for the decay factor, i.e., for the probability d the random walker to perform a random jump. A large d value favors the highly ranked documents and thus their final score, while a small value favors the connected entities which in turn favors the documents associated with highly scored entities. Note that for $d = 1.0$, the connectivity of the graph nodes is not considered (the detected entities are not taken into consideration), which means that the scoring is affected only by the random jumps, i.e., by the document scores. Since we want to favor documents associated with important (highly scored) entities we can define a small value, e.g., $d < 0.4$.

Regarding p_3, i.e., the probability to select a document-node or a related entity/property node from an entity-node, and as we have already stated, we consider that when the walker is in an entity-node, it is more probable to move to a document node (that contains/refers this entity) since the final user target is to locate documents that satisfy her/his information need. Thereby, we can define a big p_3 value, e.g., $p_3 > 0.6$. In case we want to allow only the selection of document nodes, we can define $p_3 = 1.0$.

Initial PageRank values. The algorithm requires some initial values for the graph nodes. We define a uniform distribution, specifically $1/|\mathcal{E}|$ ($\mathcal{E}$ is the set of STG nodes).

Number of iterations. According to [Page et al., 1999], the number of iterations required for convergence is empirically O(log$n$), where n is the number of edges.

Exploiting the Outcome

After running the above algorithm, all graph nodes receive a PageRank-like score. The higher the score of a node is, the most important (and relevant to the search context) that node is considered. Documents with important (highly-scored) entities will receive a high score. The documents are finally re-ranked according to their PageRank-like scores and the new list of documents is returned to the user.

The proposed re-ranking method can be exploited by any search system (directly or on-demand) operating over a collection of biomedical documents. The input is a ranked list of results and the output is the same list re-ranked. In addition, the result of this analysis can be exploited in other contexts. For example, a search system can offer entity-based faceted exploration of the search results [Fafalios et al., 2012, Fafalios and Tzitzikas, 2014a]. In this case, a facet corresponds to a category of entities, while the

entities in a facet can be ordered according to their final PageRank-like scores. Such a functionality allows the user to browse for results associated with one or more entities. For instance, in a medicine-related search application, the user can quickly locate results containing information about a specific disease, drug, etc.

Experimental Evaluation

Corpus and Setup

We used the dataset (documents, topics, relevance judgements) provided by the TREC Clinical Decision Support (CDS) track of 2014 and 2015 [Roberts et al., 2015]. The CSD track focuses on retrieving biomedical articles (publications) relevant for answering generic clinical questions about medical records (topics).

Corpus. The collection is a snapshot of the Open Access Subset of PubMed Central (PMC) containing 733,138 articles (obtained on January 21, 2014). PMC is an online digital database of freely available full-text biomedical literature. The full text of each article is represented as an NXML file (XML encoded using the NLM Journal Archiving and Interchange Tag Library). For each article, the NXML file contains information like the article's title, abstract, main body, and authors, as well as metadata like publication date, publisher, journal, authors' affiliations, IDs, etc. In this evaluation, we exploit only the title, the abstract and the main body of each article.

Queries. We used the description of each of the 60 topics provided by the TREC CDS tracks (of 2014 and 2015) for querying the collection. A topic is actually a medical case narrative serving as an idealized representation of an actual medical record and it describes information such as a patient's medical history, the patient's current symptoms, tests performed by a physician to diagnose the patient's condition, the patient's eventual diagnosis as well as the steps taken by a physician to treat the patient. The following text in an example of a medical case narrative: *"A 58-year-old nonsmoker white female with mild exertional dyspnea and occasional cough is found to have a left lung mass on chest x-ray. She is otherwise asymptomatic. A neurologic examination is unremarkable, but a CT scan of the head shows a solitary mass in the right frontal lobe."* For each such narrative, an effective IR system must find documents that can help the physician to answer common generic clinical questions related to the narrative, such as what is the patient's diagnosis or what tests should the patient receive based on the medical report. By using the whole medical case narrative for querying the collection, we somehow simulate the process in which a physician receives such a medical record and uploads it to a search system for finding articles that can help him making a diagnosis, suggesting tests, etc.

Baselines. We used Apache Lucene 4.10.3 for indexing the collection (using its Standard Analyzer which finds word boundaries, downcases the words, and filters out stopwords) and we indexed the title, the abstract and the body of each document. As regards the retrieval models, we used two different models: the first is Lucene's default scoring scheme which uses a combination of the Vector Space Model and the Boolean model (`VSM`), and the second is Okapi BM25 which applies a probabilistic method (`BM25`). We selected to use these two baselines in our experiments because they are classic and widely applicable. Note that our focus is to improve a list of retrieved results when IR has not performed very well, i.e., when the list of results contains some relevant hits only in the top positions of the answer (besides, a perfect list does not need any improvement). Note that IR can fail due to several reasons. For example, the user may have not accurately described her/his information need (this is very common in exploratory search needs). For this reason, we use a widely applicable search system (Lucene) and two popular and widely applicable retrieval models without paying particular attention to the used queries.

Entity extraction. For extracting entities from the indexed fields of top retrieved documents we used `X-Link` [Fafalios et al., 2015,Fafalios et al., 2014a]. `X-Link` is a configurable, LOD-based entity extraction system which is capable to identity EoI in a document, link the detected entities with semantic resources, and enrich them with additional semantic information coming from external SKBs. We used diseases, drugs, proteins, and chemical substances (of DBpedia) as the EoI. Regarding the weight of each indexed document part, we give 0.5 to the *title*, 0.3 to the *abstract*, and 0.2 to the *body* (this setting empirically provides better results).

Testing parameters. We run experiments for different number of retrieved results (L = 100, 250, 500 and 1000), for two different decay factor values (d = 0.0 and 0.2) for the random jumps, and without

entity enrichment (i.e., p_3= 1.0). Note that a big decay factor value (i.e., big probability of random jumps to document-nodes) does not make sense because in this case the ranking will be mainly affected by the document scores/ranks and thereby the re-ranked and the initial lists are expected to be quite similar. For this reason, we examine only two small decay factor values (0.0 and 0.2). Recall that for $d = 0.0$, we do not allow random jumps to document nodes. Thus the scores are affected only by the associations between documents and extracted entities and by the corresponding transition probabilities. We compared the following lists of top-100 results:

- `BEFORE`: Initial top-100 list returned by the baseline
- `AFTER-d0`: Top-100 list after applying the proposed re-ranking approach with $d = 0.0$
- `AFTER-d2`: Top-100 list after applying the proposed re-ranking approach with $d = 0.2$
- `RANDOM`: Top-100 list after shuffling randomly the initial list returned by the baseline

We also tested the case of entity enrichment for three different p_3 probabilities (0.25, 0.5, 0.75) by enriching the entities with the property `dct:subject` from DBpedia (this property seems to provide useful information for the specified EoI). Finally, we examined the effect of each category of EoI by running experiments using one category each time.

Evaluation metrics. For evaluating the results, we used the following metrics which have been specially designed for evaluation environments with incomplete relevance data:

- `bpref` [Buckley and Voorhees, 2004]. This metric is highly correlated with average precision when full relevance assessments are available and is more robust when the relevance assessments are reduced.
- `AveP'`: Average Precision based on a condensed list (after removing all unjudged docs) [Sakai, 2007].
- `nDCG'`: Normalized Discounted Cumulative Gain based on a condensed list [Sakai, 2007]. This metric uses a graded relevance scale and actually measures the usefulness, or gain, of a document based on its position in the list of results.
- `Q'`: Q-Measure on a condensed list [Sakai, 2007]. This metric is highly correlated with average precision and its discriminative power is known to be at least as high as that of average precision.
- `rpref_relative2` [Sakai, 2007]. This metric is an alternative to `bpref` designed to handle grade relevance and uses relative normalization to emphasize misplacement penalties based on highly ranked relevant documents.
- `P@10'`: Precision at rank 10 based on a condensed list. This metric favors condensed lists containing many judged relevant documents before the judged non-relevant documents early in the ranked list.

Note that we cannot use the inferred metrics `infAP` [Yilmaz and Aslam, 2006] and `infNDCG` [Yilmaz et al., 2008] because these metrics require knowledge of all pooled documents.

Results

At first, on average `BM25` performed better than `SVM` in the top-100 `BEFORE` list, in all metrics apart `Q'` (+6% in `bpref`, almost the same in `AveP'`, +25% in `nDCG'`, -4% in `Q'`, +1% in `rpref_relative2`, +15% in `P@10'`). Note that `Q'` gives higher emphasis and penalizes the appearance of non-relevant documents. Specifically, `BM25` returned 13 relevant hits, 40 non-relevant and 47 unjudged hits on average, while `SVM` returned 11 relevant hits, 36 non-relevant and 53 unjudged hits. We notice that both retrieval models did not manage to retrieve many relevant-for-sure documents in the top-100 lists (although the number of unjudged documents is big in both cases). This enforces the need for an effective re-ranking approach that can bring these few relevant-for-sure hits in higher positions in the returned ranked list. Moreover, in case more than 100 results are retrieved and analyzed, an effective re-ranking approach could promote in the top-100 list relevant but very low-ranked documents (in positions > 100).

Figures 9 and 10 depict the average results for all 60 topics, while Tables 1 and 2 show the corresponding precise values. The presence of symbol ‡ means that the corresponding increment is statistically significant (paired t-test, α-level = 5%). We notice that for both `SVM` and `BM25`, as well as for all metrics and numbers of retrieved results, the top-100 lists are notably improved when the proposed re-ranking approach is applied for both $d = 0.0$ and $d = 0.2$ (and the improvement is statistically significant for the majority of cases). For instance, for $d = 0.2$ and $L = 500$, the average increment in the case of `SVM` is about 40% in `bpref`, 21% in `AveP'`, 24% in `nDCG'`, 18% in `Q'`, 22% in `rpref_relative2`, and 21% in `P@10'`, while for `BM25` is about 38% in `bpref`, 28% in `AveP'`, 14% in `nDCG'`, 27% in `Q'`, 28% in `rpref_relative2`, and 16% in `P@10'`. This illustrates that the proposed method moved relevant hits in higher positions in the top-100 list. Furthermore, and for the cases where $L > 100$, re-ranking promoted in the top-100 list relevant hits which though had been ranked in positions > 100 in the `BEFORE` list.

Table 1: Comparative evaluation results using SVM as baseline.

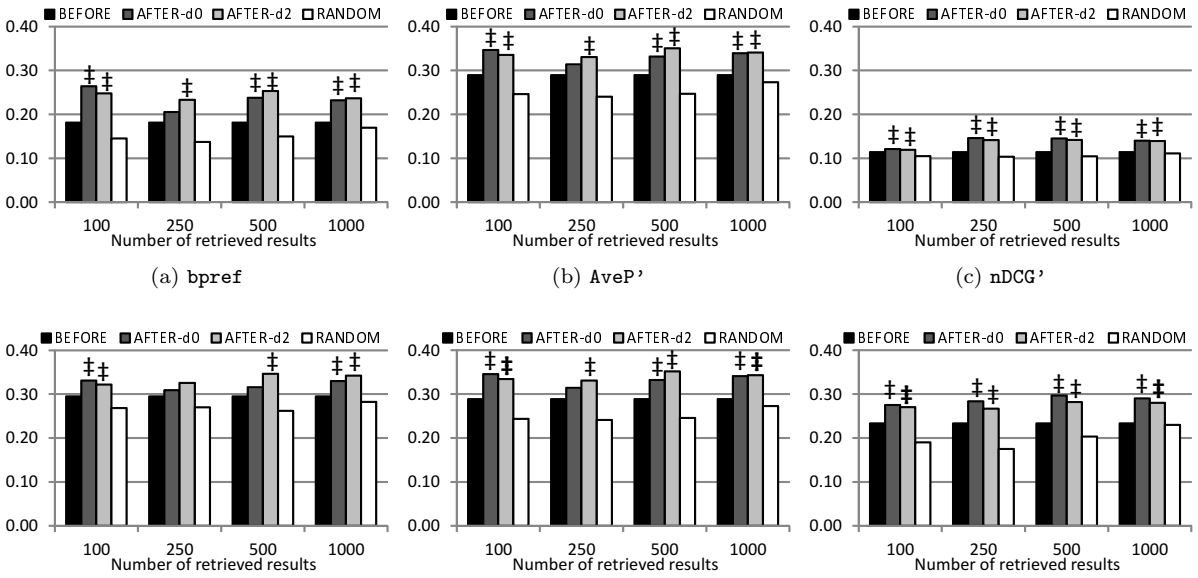
L	bpref				AveP'			
	BEFORE	AFTER-d0	AFTER-d2	RANDOM	BEFORE	AFTER-d0	AFTER-d2	RANDOM
100	0.18	0.26 ‡	0.25 ‡	0.15	0.29	0.35 ‡	0.34 ‡	0.25
250	0.18	0.21	0.23 ‡	0.14	0.29	0.31	0.33 ‡	0.24
500	0.18	0.24 ‡	0.25 ‡	0.15	0.29	0.33 ‡	0.35 ‡	0.25
1,000	0.18	0.23 ‡	0.24 ‡	0.17	0.29	0.34 ‡	0.34 ‡	0.27
L	nDCG'				Q'			
	BEFORE	AFTER-d0	AFTER-d2	RANDOM	BEFORE	AFTER-d0	AFTER-d2	RANDOM
100	0.11	0.12 ‡	0.12 ‡	0.11	0.30	0.33 ‡	0.32 ‡	0.27
250	0.11	0.15 ‡	0.14 ‡	0.10	0.30	0.31	0.33	0.27
500	0.11	0.15 ‡	0.14 ‡	0.10	0.30	0.32	0.35 ‡	0.26
1,000	0.11	0.14 ‡	0.14 ‡	0.11	0.30	0.33 ‡	0.34 ‡	0.28
L	rpref_relative2				P@10'			
	BEFORE	AFTER-d0	AFTER-d2	RANDOM	BEFORE	AFTER-d0	AFTER-d2	RANDOM
100	0.29	0.35 ‡	0.33 ‡	0.24	0.23	0.28 ‡	0.27 ‡	0.19
250	0.29	0.31	0.33 ‡	0.24	0.23	0.28 ‡	0.27 ‡	0.17
500	0.29	0.33 ‡	0.35 ‡	0.25	0.23	0.30 ‡	0.28 ‡	0.20
1,000	0.29	0.34 ‡	0.34 ‡	0.27	0.23	0.29 ‡	0.28 ‡	0.23

Table 2: Comparative evaluation results using BM25 as baseline.

L	bpref				AveP'			
	BEFORE	AFTER-d0	AFTER-d2	RANDOM	BEFORE	AFTER-d0	AFTER-d2	RANDOM
100	0.19	0.26 ‡	0.25 ‡	0.17	0.29	0.36 ‡	0.35 ‡	0.28
250	0.19	0.26 ‡	0.24 ‡	0.15	0.29	0.36 ‡	0.35 ‡	0.27
500	0.19	0.26 ‡	0.26 ‡	0.18	0.29	0.37 ‡	0.37 ‡	0.28
1,000	0.19	0.22	0.24	0.17	0.29	0.35 ‡	0.35 ‡	0.27
L	nDCG'				Q'			
	BEFORE	AFTER-d0	AFTER-d2	RANDOM	BEFORE	AFTER-d0	AFTER-d2	RANDOM
100	0.14	0.15 ‡	0.15 ‡	0.13	0.28	0.33 ‡	0.32 ‡	0.30
250	0.14	0.17 ‡	0.17 ‡	0.13	0.28	0.34 ‡	0.33 ‡	0.29
500	0.14	0.16 ‡	0.16 ‡	0.14	0.28	0.37 ‡	0.36 ‡	0.29
1,000	0.14	0.16	0.16	0.13	0.28	0.35 ‡	0.34 ‡	0.28
L	rpref_relative2				P@10'			
	BEFORE	AFTER-d0	AFTER-d2	RANDOM	BEFORE	AFTER-d0	AFTER-d2	RANDOM
100	0.29	0.36 ‡	0.35 ‡	0.28	0.27	0.30 ‡	0.30 ‡	0.22
250	0.29	0.36 ‡	0.35 ‡	0.27	0.27	0.31 ‡	0.31 ‡	0.22
500	0.29	0.37 ‡	0.37 ‡	0.28	0.27	0.31 ‡	0.31 ‡	0.24
1,000	0.29	0.35 ‡	0.35 ‡	0.27	0.27	0.31	0.31 ‡	0.22

For example, and as regards the latter, in case of SVM and for $d = 0.2$ and $L = 500$ the number of relevant-for-sure hits in the top-100 list was increased for 43/60 topics (+1 for 13 topics, +2 for 6 topics, + >2 for 24 topics), was decreased for 7/60 topics (−1 for 5 topics, −2 for 1 topic, −3 for 1 topic), and remained the same for 10/60 topics. We notice also that in all cases the random lists are worse than the initial lists. This somehow illustrates the non-randomness of our results. In addition, it is interesting that the improvement is higher for BM25 for the majority of cases (recall that BM25 performed better than SVM). This is due to the fact that, for BM25, more relevant hits are returned in the top positions of the initial list. Since entities identified in the top positions are given a high *importance score*, low ranked hits containing these entities are promoted in higher positions. Finally, as regards the decay factor, we cannot conclude safe results for the case of SVM. However, for BM25 which provides more statistically significant improvements, it seems that $d = 0.0$ performs better for the majority of cases (18/24).

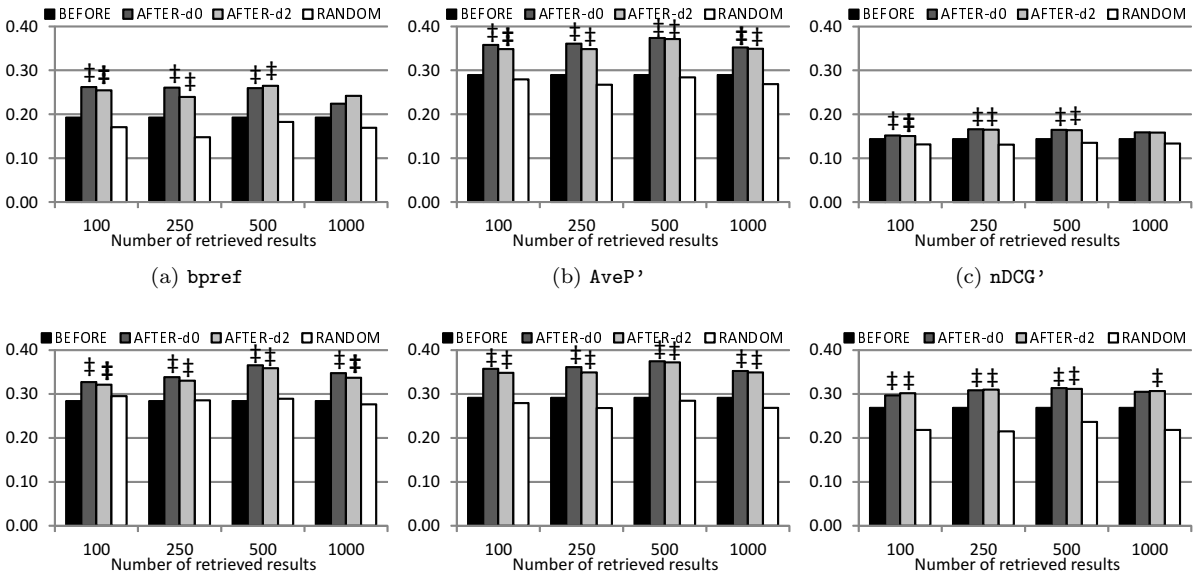
Improvement failure. Although we saw that, on average, re-ranking improves the top-100 lists, by analyzing thoroughly the results for the case of SVM, for $d = 0.2$ and $L = 500$ and for one of the evaluation metrics (bpref), we noticed that for 10 topics re-ranking failed to improve the top-100 list. By inspecting these cases, we noticed that for 7/10 topics the initial number of relevant-for-sure hits was above the average value. Moreover, only for 4 topics the reduction was above 0.05 and for 3 of these 4 topics, the number of relevant hits was above 30. This means that when the number of relevant retrieved hits is high, re-ranking can affect negatively their ranking. This can happen because, for example, a non-relevant hit exists in a top position (e.g., in the top-10 list). Entities mentioned in this result will have a high



(a) `bpref` (b) `AveP'` (c) `nDCG'`

(d) `Q'` (e) `rpref_relative2` (f) `P@10'`

Figure 9: Comparative evaluation results using `SVM` as baseline.



(a) `bpref` (b) `AveP'` (c) `nDCG'`

(d) `Q'` (e) `rpref_relative2` (f) `P@10'`

Figure 10: Comparative evaluation results using `BM25` as baseline.

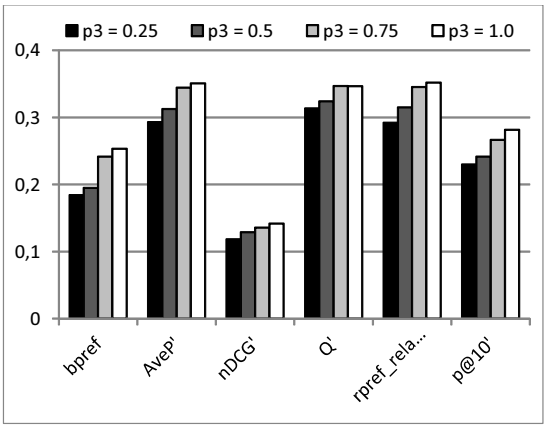


Figure 11: Effect of entity enrichment.

importance score and this will favor low ranked but probably irrelevant hits in higher positions (placing them above some of the many relevant hits that exist in the whole list, causing the decrease in precision). However, it is very interesting that, for 8/10 topics the number of relevant-for-sure hits in the top-100 list was increased (+1 for 2 topics, +2 for 1 topics, + > 2 for 5 topics), while only for 2 of these topics it was decreased (−1 for the one and −2 for the other). This means that, although re-ranking moved some highly-ranked relevant-for-sure hits in lower positions (causing the reduction of `bpref` value), it nevertheless managed to bring more relevant hits in the top-100 list, increasing thereby recall. Note that improvement in recall may be very important for search applications in professional domains where the main goal is to retrieve as many relevant documents as possible (e.g., searching for publications or patents).

By inspecting more cases, we also noticed that re-ranking fails when the results are very bad, specifically when there are no relevant-for-sure hits in the top positions. Since entities identified in theses top positions are given a high *importance score*, low ranked (but probably irrelevant) hits containing these entities are promoted in higher positions. Thereby, we can conclude that, as expected, when the initial list does not contain any relevant hits in the top-ranked positions, re-ranking fails to improve the list of results.

Effect of entity enrichment. Using `SVM` as baseline, we tested the case of entity enrichment for three different p_3 probabilities (0.25, 0.5, 0.75), keeping constant the decay factor ($d = 0.2$) and the number of retrieved results ($L = 500$), and we compared it with the no entity enrichment approach (i.e., for $p_3 = 1.0$). Figure 11 depicts the results. We notice that entity enrichment did not improve the top-100 list and this is clear for all evaluation metrics. Furthermore, the smaller the value of p_3 is (the larger the probability for the random walker to select a related entity/property node), the worse the results are. This means that the specific semantic information about the detected entities (DBpedia `subject` property), although it might be quite useful in another context (e.g., faceted search [Tzitzikas et al., 2016]), it misleads the random walker and affects negatively the re-ranking of the retrieved results. As an example, an entity with high importance score (i.e., identified in many top results) may share the `subject` property with some other entities which though are not relevant to the particular medical case. For instance, the drug *Aspirin* shares with other drugs subjects like *salicylates* (nonsteroidal anti-inflammatory drugs), *antiplatelet drugs*, *acetate esters*, and *German inventions*. However, such associations will give high scores to the connected entities which in turn will favor documents containing these entities. For example, regarding the category *German inventions*, documents mentioning some other drugs invented in Germany will be promoted in higher positions but probably these documents will not be relevant.

Effect of category of entities. We examined the effect of each category of EoI (disease, drug, protein, and chemical substance) by running experiments using one category each time. We used `BM25` as baseline, $L = 250$ and tested both $d = 0.0$ and $d = 0.2$. We compared the results with the `BEFORE` list as well as with the list which considers all categories (`ALL`). Figures 12 and 13 depict the results for $d = 0.0$ and $d = 0.2$, respectively. We notice that the category disease provides the best results, with a performance very close to the `ALL` case, while for $d = 0.2$ and considering the metrics `Q'` and `AveP'`, it performs better than `ALL`. This means that this category of entities contributes more on the improvement of the re-ranked list. In addition, we notice that for some evaluation metrics (`nDCG'`, `P@10'`), the other three categories have a negative effect on the `BEFORE` list. The above results illustrate that a better selection of the EoI

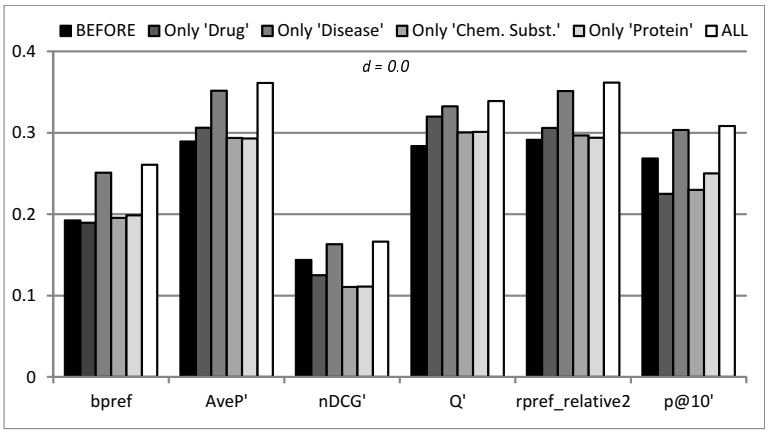


Figure 12: Effect of category of entities ($d = 0.0$).

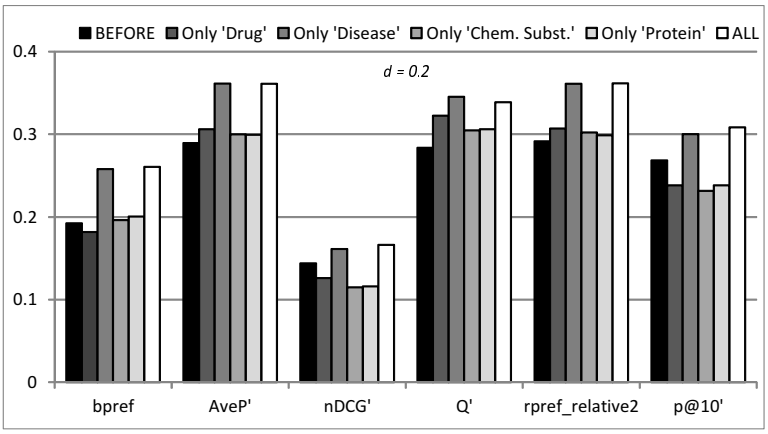


Figure 13: Effect of category of entities ($d = 0.2$).

can produce better rankings (e.g., in our case we can consider only the categories disease, drug). Studying approaches for *learning to rank* the results based on the contribution or significance of each category of entities is out of the scope of this paper but an important direction for future research.

Efficiency of the analysis process. The computational and time complexity of the entire analysis process is affected by many parameters like the number of top results we analyze, the efficiency of the entity extraction system (in case of real-time processing), and the average size of the documents in the corpus. Regarding the proposed Random Walk-based algorithm, its time complexity is *linear* to the number of graph vertices, i.e., to the number of retrieved documents and the number of extracted entities. Previous work which exploits a common algorithm for offering entity-based faceted exploration of search results has shown than the average execution time for a graph of about 4,000 vertices is less than 330 ms (using a modest personal computer with only 4 GB of main memory) [Fafalios et al., 2014b].

Concluding Remarks and Future Research

We have introduced an entity-based approach for re-ranking a list of medical search results. The objective is to improve the list of results by promoting low-ranked but relevant hits referring medical entities that are important to the current search context. The approach is based on named-entity extraction applied in a set of retrieved documents, and on a graph of documents and extracted entities that is constructed dynamically and analyzed stochastically. The proposed method is *general* (applicable over existing search systems), *configurable* (applicable also to other domains), *exploitable* also in other contexts (faceted search, query-expansion, etc.), while the process is fully automated (no user effort is required).

Experimental results over the dataset of the TREC Clinical Decision Support track using two classic and widely applicable baselines, illustrated a significant improvement of the new lists of results for the majority of queries. For instance, re-ranking the top-500 hits using the proposed approach, we can achieve about 40% better `bpref` and about 20% better `AveP'` (average precision based on condensed lists) of the top-100 list. This means that the number of relevant hits in the final top-100 list is increased and that

existing relevant hits are promoted in higher positions. However, when the number of relevant retrieved hits is very high or very low, re-ranking can affect negatively the results by promoting irrelevant hits in higher positions. Nevertheless, in the former case of big number of relevant hits, the results showed that re-ranking can further improve recall. Finally, we saw that additional semantic information about the entities (properties and related entities) can affect negatively the re-ranking process and thus must be carefully considered during the stochastic analysis.

To sum up, such a functionality can help physicians to locate faster bigger number of relevant documents, since low-ranked but relevant hits are promoted in higher positions. As regards future work and research, we plan to study how the user actions in a more interactive context (e.g., clicking on an entity in a faceted search system) can be exploited in our stochastic model (e.g., for updating the edge probabilities). It is interesting also to study approaches on how to exploit implicit user feedback (e.g., traces of user interaction in session logs) in order to understand the quality of the displayed list of results [Agichtein et al., 2006] and decide whether to automatically apply or not the proposed re-ranking method.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the BlueBRIDGE project (Grant agreement No 675680).

References

[Agichtein et al., 2006] Agichtein, E., Brill, E., and Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM.

[Babashzadeh et al., 2013] Babashzadeh, A., Huang, J., and Daoud, M. (2013). Exploiting semantics for improving clinical information retrieval. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 801–804. ACM.

[Buckley and Voorhees, 2004] Buckley, C. and Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32. ACM.

[Cao et al., 2008] Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250. ACM.

[Carpineto and Romano, 2012] Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1.

[Chang et al., 2006] Chang, Y., Ounis, I., and Kim, M. (2006). Query reformulation using automatically generated query concepts from a document space. *Information processing & management*, 42(2):453–468.

[Chidlovskii et al., 2000] Chidlovskii, B., Glance, N. S., and Grasso, M. A. (2000). Collaborative reranking of search results. In *Proc. AAAI-2000 Workshop on AI for Web Search*.

[Fafalios et al., 2014a] Fafalios, P., Baritakis, M., and Tzitzikas, Y. (2014a). Configuring named entity extraction through real-time exploitation of linked data. In *4th International Conference on Web Intelligence, Mining and Semantics (WIMS'14)*. ACM.

[Fafalios et al., 2015] Fafalios, P., Baritakis, M., and Tzitzikas, Y. (2015). Exploiting Linked Data for Open and Configurable Named Entity Extraction. *International Journal on Artificial Intelligence Tools*, 24(02).

[Fafalios et al., 2012] Fafalios, P., Kitsos, I., Marketakis, Y., Baldassarre, C., Salampasis, M., and Tzitzikas, Y. (2012). Web searching with entity mining at query time. In *5th Information Retrieval Facility Conference*.

[Fafalios et al., 2014b] Fafalios, P., Papadakos, P., and Tzitzikas, Y. (2014b). Enriching textual search results at query time using entity mining, linked data and link analysis. *International Journal of Semantic Computing*, 08(04).

[Fafalios and Tzitzikas, 2014a] Fafalios, P. and Tzitzikas, Y. (2014a). Exploratory Professional Search through Semantic Post-Analysis of Search Results. In *Professional Search in the Modern World*, volume 8830. Springer.

[Fafalios and Tzitzikas, 2014b] Fafalios, P. and Tzitzikas, Y. (2014b). Post-analysis of keyword-based search results using entity mining, linked data and link analysis at query time. In *2014 IEEE Eighth International Conference on Semantic Computing (ICSC 2014)*, Newport Beach, California, USA. IEEE.

[Goeuriot et al., 2014] Goeuriot, L., Kelly, L., Jones, G. J., Müller, H., and Zobel, J. (2014). Report on the sigir 2014 workshop on medical information retrieval (medir). In *ACM SIGIR Forum*, volume 48, pages 78–82. ACM.

[Heath and Bizer, 2011] Heath, T. and Bizer, C. (2011). Linked Data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.

[Kanhabua and Nørvåg, 2010] Kanhabua, N. and Nørvåg, K. (2010). Determining time of queries for re-ranking search results. In *Research and Advanced Technology for Digital Libraries*, pages 261–272. Springer.

[Lee et al., 2008] Lee, K. S., Croft, W. B., and Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 235–242. ACM.

[Li et al., 2015] Li, J., Liu, C., Liu, B., Mao, R., Wang, Y., Chen, S., Yang, J.-J., Pan, H., and Wang, Q. (2015). Diversity-aware retrieval of medical records. *Computers in Industry*, 69:81–91.

[Oh and Jung, 2015] Oh, H.-S. and Jung, Y. (2015). Cluster-based query expansion using external collections in medical information retrieval. *Journal of biomedical informatics*, 58:70–79.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web.

[Palotti et al., 2015] Palotti, J., Zuccon, G., Goeuriot, L., Kelly, L., Hanbury, A., Jones, G., Lupu, M., and Pecina, P. (2015). Clef ehealth evaluation lab 2015, task 2: Retrieving information about medical symptoms. In *Proc. of CLEF*.

[Park and Ramamohanarao, 2007] Park, L. A. and Ramamohanarao, K. (2007). Query expansion using a collection dependent probabilistic latent semantic thesaurus. In *Advances in Knowledge Discovery and Data Mining*. Springer.

[Riezler et al., 2007] Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., and Liu, Y. (2007). Statistical machine translation for query expansion in answer retrieval. In *Annual Meeting Association For Computational Linguistics*.

[Roberts et al., 2015] Roberts, K., Simpson, M. S., Voorhees, E., and Hersh, W. R. (2015). Overview of the trec 2015 clinical decision support track.

[Sakai, 2007] Sakai, T. (2007). Alternatives to bpref. In *30th international ACM SIGIR conference on Research and development in information retrieval*. ACM.

[Singhal, 2012] Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official google blog*.

[Sun et al., 2006] Sun, R., Ong, C.-H., and Chua, T.-S. (2006). Mining dependency relations for query expansion in passage retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

[Tao and Zhai, 2006] Tao, T. and Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169. ACM.

[Tzitzikas et al., 2016] Tzitzikas, Y., Manolis, N., and Papadakos, P. (2016). Faceted exploration of rdf/s datasets: a survey. *Journal of Intelligent Information Systems*, pages 1–36.

[Xu et al., 2009] Xu, Y., Jones, G. J., and Wang, B. (2009). Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66. ACM.

[Yilmaz and Aslam, 2006] Yilmaz, E. and Aslam, J. A. (2006). Estimating average precision with incomplete and imperfect judgments. In *15th ACM international conference on Information and knowledge management*. ACM.

[Yilmaz et al., 2008] Yilmaz, E., Kanoulas, E., and Aslam, J. A. (2008). A simple and efficient sampling method for estimating ap and ndcg. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 603–610. ACM.

[Yin et al., 2010] Yin, X., Huang, X., and Li, Z. (2010). Promoting ranking diversity for biomedical information retrieval using wikipedia. In *European Conference on Information Retrieval*, pages 495–507. Springer.

[Zhuang and Cucerzan, 2006] Zhuang, Z. and Cucerzan, S. (2006). Re-ranking search results using query logs. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 860–861. ACM.