# Entity-based Stochastic Analysis of Search Results for Query Expansion and Results Re-Ranking

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

In this paper we introduce a method for exploiting *entities* from the emerging Web of Data for enhancing various Information Retrieval (IR) services. The approach is based on named-entity recognition applied in a set of search results, and on a graph of documents and identified entities that is constructed dynamically and analyzed stochastically using a *Random Walk* method. The result of this analysis is exploited in two different contexts: for *automatic query expansion* and for *re-ranking* a set of retrieved results. Evaluation results in the 2015 TREC Clinical Decision Support Track illustrate that query expansion can increase recall by retrieving more relevant hits, while re-ranking can notably improve the ranked list of results by moving relevant but low-ranked hits in higher positions.

## 1. INTRODUCTION

The Web has now evolved to an information space where both unstructured documents (e.g. Web pages) and structured data (e.g. Linked Open Data (LOD) [8]) coexist in various forms. An important observation is that entity names (like names of persons, locations, etc.) occur in all kinds of artifacts: Web pages, database cells, RDF triples, etc. A generic hypothesis that we investigate is whether and how we can exploit named-entities for integrating documents (actually search results) with data and knowledge. The idea is to construct dynamically a graph of documents and entities, and then to analyze it stochastically using a Random Walk-based method. Specifically, we model the search process as a random walker of the graph defined by the top documents returned by a search system and the entities identified in these documents. For analyzing the graph and scoring its nodes, we exploit both the ranking of the returned search hits, the "importance" (within the search context) of the extracted entities and their connectivity. The result of this analysis is exploited in two different contexts:

- for automatic *query expansion*, aiming to construct and submit a new query that can retrieve more relevant hits
- for *re-ranking* the set of retrieved results, aiming to promote low-ranked but relevant hits referring important (for the search context) entities

Figure 1 depicts the steps of the analysis process. At first, the user submits a query to a search system and the top-L (e.g. $L = 1,000$) results are retrieved. Then, Named Entity Recognition (NER) is applied in these results for identifying LOD entities. In the next (optional) step, more semantic information about the identified entities is retrieved from the LOD (like properties and related entities). A graph of search results and (semantically-enriched) entities is constructed and analyzed stochastically. The document and entity scores (given by the probabilistic analysis) are exploited for enhancing various IR services, e.g. for query expansion and results re-ranking. Finally, the user interacts with the results.

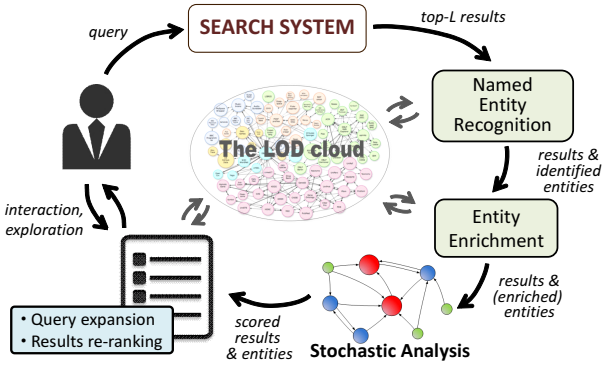


Figure 1: The steps of the analysis process.

We evaluated the effectiveness of the proposed method in the medicine domain, in the context of the 2015 TREC Clinical Decision Support Track. The results showed that: (i) query expansion increases the number of relevant hits for the majority of topics (about +70% in average) (ii) re-ranking improves the ranked list of results returned by a classical IR system for the majority of topics (about +33% average increment in `infNDCG` and +47% in `P@10`) (iii) additional semantic information about the entities (properties and related entities) can affect negatively the re-ranking process.

**Related Work.** The work in [4,6] introduced an exploratory search process based on extracted entities in which the search results are connected with data and knowledge at query time with no human effort. To make such a service feasible for large amounts of data, [9] details a distributed computation

model and shows how the required computational tasks can be factorized and expressed as MapReduce functions. For identifying the semantic information (entities and properties) that better characterizes the search results, the work in [5, 7] introduces a Random Walk-based ranking model that exploits both the ranking of the returned results, the extracted named entities and their connectivity, and which is exploited for producing and showing to the user top-K semantic graphs related to the search results. In this work, we continue this line of research and we investigate whether and how such "overview" information (entities detected in the search results and semantic information associated to these entities) can be exploited for re-ranking a list of results as well as for query-expansion.

## 2. STOCHASTIC ANALYSIS

**Query and retrieved documents.** At first, the user submits a query $q$ describing an information need to a search system. Let $L$ be the number of top documents (hits) to retrieve, $A$ the set of these top-$L$ documents, and $score(a)$ the score (value in the range $[0, 1]$) of a document $a \in A$. Moreover, let $P$ be 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*), and $w(p)$ the weight expressing the importance of a part $p \in P$, where $\sum_{p' \in P} w(p') = 1$.

**Extracted entities.** Now, a LOD-based NER system (e.g. X-Link [3] or DBpedia Spotlight [11]) is exploited for identifying entities of interest (names of entities that are important in the application context) in all parts of each retrieved document. A list of identified entities is derived. Each entity is accompanied by its URI in a semantic knowledge base, e.g. DBpedia [10]. In more details, let $ent(a_p)$ be the set of entities identified in the part $p$ of a document $a \in A$, and $ent(a) = \cup_{p \in P} ent(a_p)$ the set of all entities identified in a document $a$ (in all its parts). $E = \cup_{a \in A} ent(a)$ is the set of all entities identified in $A$. Conversely, let $docs(e) = \{\ a \in A \mid e \in ent(a)\}$ be the elements of $A$ in which $e$ has been identified. Finally, let $ef(e, a_p)$ be the frequency (number of occurrences) of the entity $e$ in the part $p$ of the document $a$.

The *importance* of an entity $e$ identified in a document $a$ is defined as: $imp(e, a) = \sum_{p \in P} (\frac{ef(e,a_p)}{\max_{e' \in ent(a_p)} ef(e',a_p)} \cdot w(p))$. The score takes into account both the number of entity occurrences and the part(s) in which the entity has been identified. In the entire set of top-L documents, the importance of an entity $e$ is defined as:

$$ImpScore(e) = \sum_{a \in docs(e)} (imp(e, a) \cdot score(a)) \qquad (1)$$

We notice that the score is higher if $e$ has been identified in the top scored documents. This entity importance score is actually a variation of the formula proposed in [4] for scoring entities identified in a set of search results.

**Graph construction.** We first construct a semantic graph of documents and entities, denoted by $\mathcal{X}$. We consider both the documents and the entities as vertices in $\mathcal{X}$, while for drawing the edges we take into account in which documents an entity was identified. 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 identified in $a$. Now, by exploiting a semantic knowledge base, we fetch interesting (for the search context) triples that describe information about the identified entities, like properties and related entities (recall that each identified entity is accompanied by its URI in a semantic knowledge base). We enrich the graph with the corresponding properties, entities and associations. Let $R$ be this set of related properties and entities (not identified in the search results).

Now we transform the graph to a State Transition Graph (STG), denoted by $\mathcal{G} = (\mathcal{E}, \mathcal{P})$. We do that by simply considering also the opposite direction for each directed edge. 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 ($e$, *"identifiedIn"*, $a$) or that ($a$, *"contains"*, $e$), i.e. the difference lies in how we name the property.

**Edge Weighting.** For weighting the edges, we consider the document and entity scores. Specifically, the edge weight from a node $n'$ to a connected node $n$ is defined as:

$$weight(n' \to n) = \begin{cases} \frac{ImpScore(n)}{\sum_{e' \in ent(n')} ImpScore(e')} & n' \in A,\ n \in E \\ p \cdot \frac{score(n)}{\sum_{a' \in A} score(a')} & n' \in E,\ n \in A \\ (1-p) \cdot \frac{1}{|out(n')|} & n' \in E,\ n \notin A \\ \frac{1}{|out(n')|} & n' \in R,\ n \in E \end{cases} \qquad (2)$$

where $p$ is the probability the random walker to select a document-node when being at an entity-node (note that the weights of the outgoing edges of a single node must represent transition probabilities, i.e. they must sum to 1). We notice that when the walker lies in a document-node, the transition probabilities are affected by the importance of the connected entities. Specifically, the higher the score of an entity is, the higher is the probability to move to that entity. Similarly, in case the walker lies in an entity-node, the transition probabilities to document-nodes are affected by the document scores, while the transition probabilities to properties and related entities are defined equiprobably. Finally, when the walker lies in a related property/entity node, he can move to the connected entity-nodes equiprobably.

**Analyzing the STG.** The objective is to find the probability the random walker to be in a specific node. We analyze the STG using a PageRank-like scoring formula. 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:

$$r(n) = d \cdot Jump(n) + (1-d) \sum_{n' \in in(n)} (weight(n' \to n) \cdot r(n')) \quad (3)$$

where $d$ is the probability (decay factor) the walker to perform a random jump, $Jump(n)$ expresses the probability the walker to jump to the node $n$, and $weight(n' \to n)$ is the probability (as defined in Formula 2) the walker to visit $n$ when being in a node $n'$ connected to $n$. As regards $Jump(n)$, we allow the random jumps only to nodes

corresponding to documents, i.e. $Jump(n) = 0$ if $n \notin A$. In addition, we adjust the jump probabilities according to the document scores (instead of assuming a uniform distribution). Specifically, for a node $n \in A$ we consider the following formula for the random jumps:

$$Jump(n) = \frac{score(n)}{\sum_{a' \in A} score(a')} \quad (4)$$

PageRank requires some initial values for the graph nodes. We can define a uniform distribution. Specifically, for each node $n$ we set $r(n) = 1/|\mathcal{E}|$ ($\mathcal{E}$ is the set of STG nodes). Finally, the values are computed iteratively and iterations should be run to convergence. According to [12], 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 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.

*Query Expansion.* We exploit the top-scored entities (e.g. the top-10) for expanding the query string. Note that an identified entity may actually be a synonym (or scientific name) of a term in the query, or in the case of medical records, a case narrative may actually correspond to a particular disease, thus including this entity name in the query string may invoke the retrieval of new relevant documents (that did not exist in the initial list of results), while some relevant documents may be moved in higher ranks.

*Re-ranking the Retrieved Results.* All document-nodes have received a final PageRank score. We exploit these scores for re-ranking the list of retrieved results. Low-ranked documents referring highly-scored entities will now receive a high score and will be promoted in the new ranked list of results.

# 3. EVALUATION

## 3.1 Corpus and Setup

We evaluated the proposed approach in the 2015 TREC Clinical Decision Support track[1]. The track focuses on retrieving biomedical articles relevant for answering generic clinical questions about medical records. We used Apache Lucene[2] for indexing the collection while we indexed the *title*, the *abstract* and the *body* of each document.

As regards the topics, each one is a medical case narrative serving as an idealized representation of an actual medical record and it describes information such as the patient's medical history, current symptoms, etc. For each provided topic, an effective IR system must find documents that can help the physician to answer a common generic clinical question such as what is the patient's diagnosis or what tests should the patient receive based on the medical report. The provided 30 topics are annotated according to the three most common generic clinical question types [1]: *diagnosis* (what is the patient's diagnosis based on the medical report), *test* (what tests should the patient receive based on the medical report), *treatment* (how should the patient be treated based on the medical report). The first 10 topics are of type *diagnosis*, the next 10 topics are of type *test*, while the last 10 are of type *treatment*. Finally, for each topic a description and a (smaller) summary is given.

For performing NER in the indexed fields of top retrieved documents we used `X-Link`. `X-Link` [2, 3] is a configurable, LOD-based NER system capable to identity entities in a document, link the identified entities with semantic resources, and enrich them with additional semantic information coming from external semantic knowledge bases. As the entities of interest, we used diseases, drugs, proteins, and chemical substances coming from DBpedia, while for testing the case of entity enrichment, we used the DBpedia `dct:subject` property[3].

## 3.2 Submitted Runs

We submitted the following 4 runs for testing the proposed *re-ranking* approach using different parameters:

- `(RRd0)` Result re-ranking with topic *description* as the query, $L = 250$, $d = 0.0$, $p = 1.0$
- `(RRd0+EE)` Result re-ranking with topic *description* as the query, $L = 250$, $d = 0.0$, $p = 0.7$ (entity enrichment is applied)
- `(RRd2)` Result re-ranking with topic *description* as the query, $L = 250$, $d = 0.2$, $p = 1.0$
- `(RRs0)` Result re-ranking with topic *summary* as the query, $L = 250$, $d = 0.0$, $p = 1.0$

We submitted 1 run for testing *query expansion* on the *re-ranked* list returned by Lucene:

- `(RRd0+QE)` Query expansion with the top-10 scored entities, using topic description as the initial query, $L = 250$, $d = 0.0$, $p = 1.0$

We also submitted 1 run for testing the effect of *re-ranking* on the list returned by Lucene after *query expansion*.

- `(RRd0+QE+RRd0)` Query expansion using the top-10 entities, then stochastic re-ranking, using the topic description as the initial query, $L = 250$, $d = 0.0$, $p = 1.0$

## 3.3 Results

At first we should point out that the top-1000 initial list returned by Lucene contains in average 40 relevant-for-sure hits (almost the same, in average, using either the topic description or the topic summary as the submitted query). This means that Lucene did not manage to retrieve many relevant-for-sure documents in the top-1000 list. This enforces the need for an effective query expansion approach that can retrieve more relevant hits, or for an effective re-ranking approach that can bring these few relevant-for-sure documents in higher positions in the returned ranked list.

### 3.3.1 Query Expansion Effect

Query expansion managed to retrieve more relevant hits for the majority of topics. Specifically, the number of relevant hits was increased for 18/30 topics, was reduced for 9/30 topics, and remained the same for 3/30 topics. In average, the number of relevant hits was increased about 70% in these 18 topics. Figure 2 depicts the results for all 30 topics. We notice that for some topics the improvement is very large, e.g. for a *test* topic, query expansion managed to retrieve +87 relevant hits and for a *diagnosis* topic +61

[1] `http://trec-cds.appspot.com/`

[2] `https://lucene.apache.org/`

[3] In DBpedia, the `dct:subject` property (`http://purl.org/dc/terms/subject`) provides categories/groups in which the corresponding entity belongs.

relevant hits. Such improvement in recall may be very important for search applications in professional domains (like in the medicine domain) where the main goal is to retrieve almost all documents that are relevant to an issue.

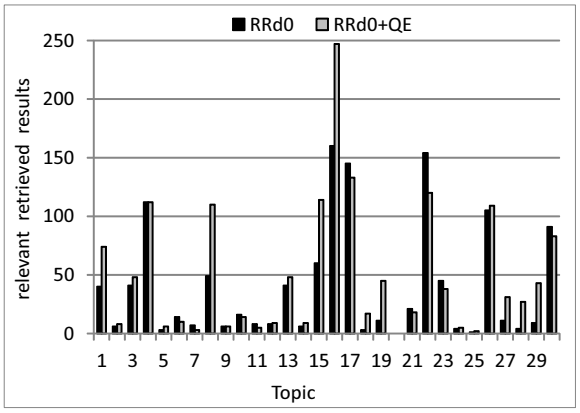


Figure 2: Query expansion effect on number of relevant retrieved results per topic

### 3.3.2 Re-Ranking Effect

As regards the effect of *re-ranking* in the list of results derived by the query expansion approach, re-ranking improved the list of results for the majority of topics. Specifically, as regards the `infNDCG` evaluation metric, the value was increased for 20/30 topics, was reduced for 7/30 topics and remained the same for 3/30 topics. As regards the `P@10` metric, the value was increased for 11/30 topics, was reduced for 2/30 topics and remained the same for 17/30 topics. For the cases in which we had improvement, `infNDCG` was increased about 33% in average, while `P@10` about 47%. These results show that the proposed re-ranking method managed to move relevant but initially low-ranked hits in higher positions for the majority of topics. Figures 3 and 4 depict the results for all topics.

### 3.3.3 Entity Enrichment Effect

Entity enrichment affected negatively the results for the majority of topics. Specifically, as regards the `infNDCG` evaluation metric the value was increased for 10/30 topics, reduced for 17/30 topics, and remained the same for 3/30 topics. As regards the `P@10` metric, the value was increased for 4/30 topics, reduced for 12/30 topics, and remained the same for 14/30 topics. This means that the specific semantic information about the identified entities (DBpedia `subject` property) can mislead the random walker and affect negatively the re-ranking of the retrieved results. Figures 5 and 6 depict the full results.

### 3.3.4 Description vs Summary

We compared the number of relevant retrieved hits when using the topic description as the submitted query and when using the topic summary. Figure 7 depicts the results. When using topic summary, the number of relevant hits is increased for 13 topics while it is reduced for 14 topics. We also notice that for some topics (for the topics 1, 4, 8, 16, 21, 22) the number of relevant hits in each approach differs significantly. Specifically, for the topics 1 and 16, description retrieved

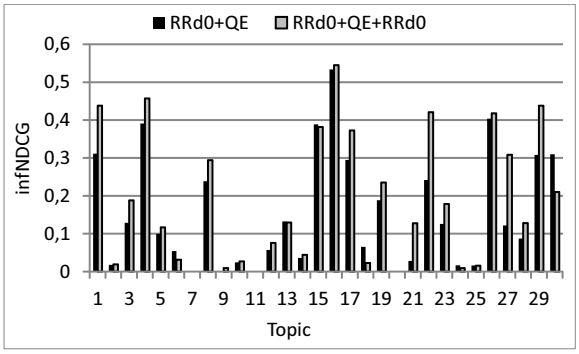


Figure 3: Re-ranking effect on infNDCG per topic

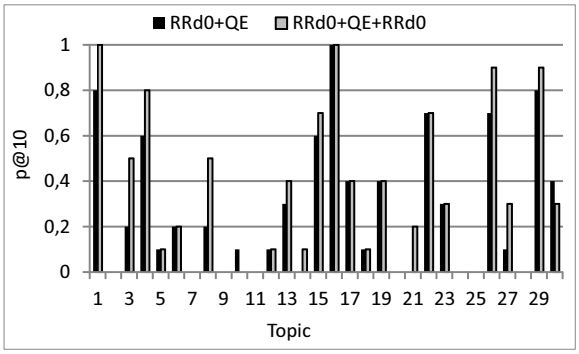


Figure 4: Re-ranking effect on P@10 per topic

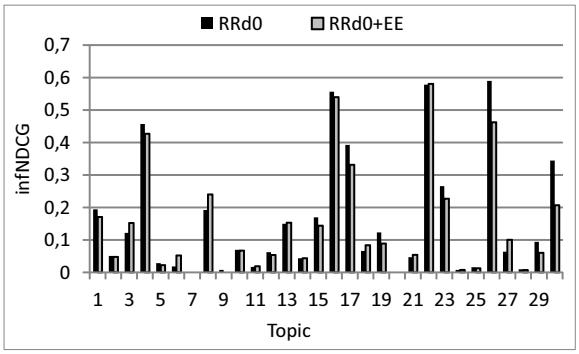


Figure 5: Entity enrichment effect on infNDCG per topic

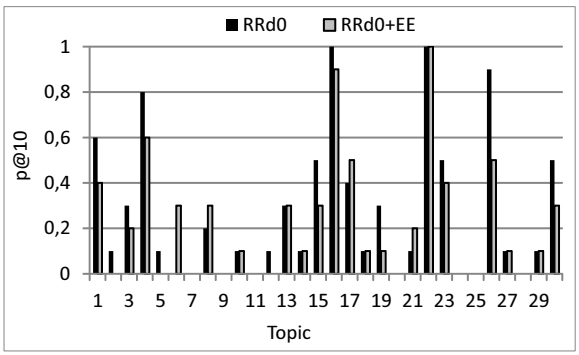


Figure 6: Entity enrichment effect on P@10 per topic

more relevant hits, while for the topics 4, 8, 21, 22 summary performed better. From the above results we cannot infer a safe conclusion about which topic part is better to use for querying the underlying search system.

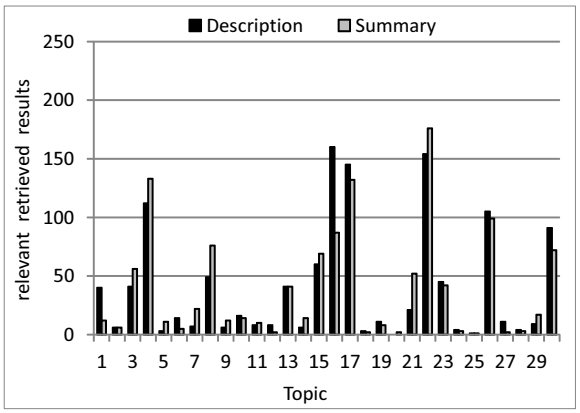


Figure 7: Submitted query: description vs summary

### 3.3.5 *Comparison with the other TREC systems*

As regards the comparison with the other systems that participated in the same TREC track, at first we should stress that, in our context, such a comparison is not important because we focus on how to *improve* an existing list of results returned by another search system, i.e. we act in a meta-search level. Nevertheless, here we present the results for the `RRd0` run (the results are similar for all runs). As regards the evaluation metric `infNDCG`, our system was above average for 11/30 topics, below average for 18/30 topics, and same as average for 1 topic. As regards the metric `P@10`, our system was above average for 10/30 topics, below average for 14/30 topics, and same as average for 6/30 topics. Figures 8 and 9 depict the results for all topics.

## 4. CONCLUSION

We have proposed a generic entity-based approach for *query expansion* and *results re-ranking*. The approach is based on named entity recognition applied in a set of retrieved documents, and on a graph of documents and entities that is constructed dynamically and analyzed stochastically. Experimental results in the 2015 TREC Clinical Decision Support Track showed that: i) query expansion can notably increase recall by retrieving more relevant hits (70% more relevant hits in our experiments), ii) re-ranking can improve the ranked list of results returned by a classical IR system by moving low-ranked but relevant hits in higher positions (30% increase in `infNDCG` and 47% in `P@10` in our experiments), and iii) additional semantic information about the entities (like properties and related entities) can affect negatively the re-ranking process and thus must be carefully considered during the stochastic analysis.

In future we plan to perform a more extensive evaluation using more topics. Our aim is to infer under what circumstances the proposed approach is effective and enhances IR, or ineffective and thus must be avoided.

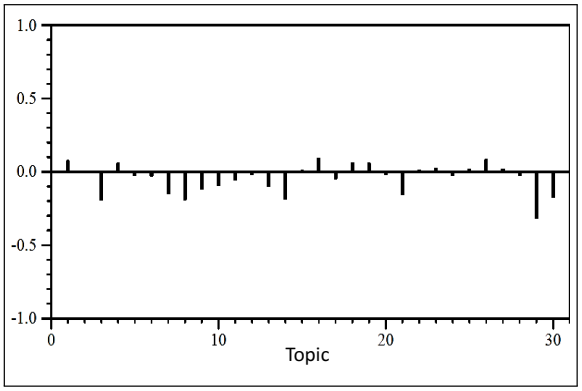


Figure 8: Difference from median `infNDCG` per topic

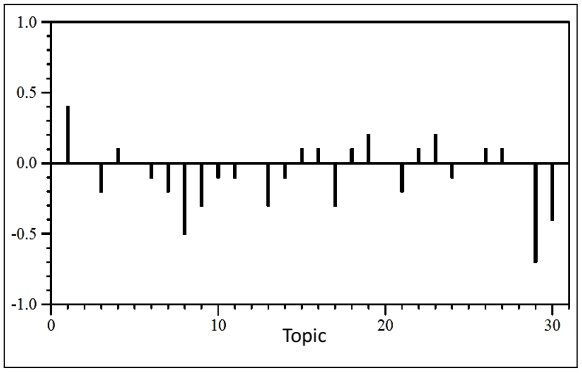


Figure 9: Difference from median `P@10` per topic

## Acknowledgements

This work was partially supported by BlueBridge (H2020 Research Infrastructures, 2015-2018).

## 5. REFERENCES

[1] J. W. Ely, J. A. Osheroff, P. N. Gorman, M. H. Ebell, M. L. Chambliss, E. A. Pifer, and P. Z. Stavri. A taxonomy of generic clinical questions: classification study. *Bmj*, 321(7258):429–432, 2000.

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

[3] P. Fafalios, M. Baritakis, and Y. Tzitzikas. Exploiting linked data for open and configurable named entity extraction. *International Journal on Artificial Intelligence Tools*, 24(02), 2015.

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

[5] P. Fafalios, P. Papadakos, and Y. Tzitzikas. Enriching textual search results at query time using entity mining, linked data and link analysis. *International Journal of Semantic Computing*, 2015.

[6] P. Fafalios and Y. Tzitzikas. Exploratory Professional Search through Semantic Post-Analysis of Search Results. In *Professional Search in the Modern World*,

volume 8830 of *Lecture Notes in Computer Science*, pages 166–192. Springer, 2014.

[7] P. Fafalios and Y. Tzitzikas. 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)*, 2014.

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

[9] I. Kitsos, K. Magoutis, and Y. Tzitzikas. Scalable entity-based summarization of web search results using mapreduce. *Distributed and Parallel Databases*, 32(3):405–446, 2014.

[10] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 2014.

[11] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *7th International Conference on Semantic Systems*. ACM, 2011.

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