Boosting Document Retrieval with Knowledge Extraction and Linked Data

Marco Rospocher [*], Francesco Corcoglioniti and Mauro Dragoni
Fondazione Bruno Kessler, Trento, Italy

Editor: Andreas Hotho, Julius-Maximilians-Universität Würzburg, Germany
Solicited reviews: Sébastien Harispe, Ecole des Mines d'Alès, France; Harald Sack, FIZ Karlsruhe, Germany; One anonymous reviewer

Abstract. Given a document collection, Document Retrieval is the task of returning the most relevant documents for a specified user query. In this paper, we assess a document retrieval approach exploiting Linked Open Data and Knowledge Extraction techniques. Based on Natural Language Processing methods (e.g., Entity Linking, Frame Detection), knowledge extraction allows disambiguating the semantic content of queries and documents, linking it to established Linked Open Data resources (e.g., DBpedia, YAGO) from which additional semantic terms (entities, types, frames, temporal information) are imported to realize a semantic-based expansion of queries and documents. The approach, implemented in the KE4IR system, has been evaluated on different state-of-the-art datasets, on a total of 555 queries and with document collections spanning from few hundreds to more than a million of documents. The results show that the expansion with semantic content extracted from queries and documents enables consistently outperforming retrieval performances when only textual information is exploited; on a specific dataset for semantic search, KE4IR outperforms a reference ontology-based search system. The experiments also validate the feasibility of applying knowledge extraction techniques for document retrieval — i.e., processing the document collection, building the expanded index, and searching over it — on large collections (e.g., TREC WT10g).

Keywords: Information Retrieval, Document Retrieval, Knowledge Extraction, Semantic Web, Large-scale Processing

1. Introduction

Document Retrieval is a well-know Information Retrieval (IR) task consisting in returning documents relevant to a given user query from a document collection. Traditional IR approaches solve this task by computing the similarity between terms or possible term-based expansions (e.g., synonyms, related terms) of the query and the documents. These approaches tend to suffer of known limitations, that we exemplify with the query "astronomers influenced by Gauss": relevant documents may not necessarily contain all the query terms (e.g., terms "influenced" or "astronomers" may not be used at all in a relevant document); similarly, some relevant documents may be ranked lower than others containing all three terms, but in an unrelated way (e.g., a document about some astronomer, containing the information that he was born centuries before Gauss and was influenced by Leonardo Da Vinci).

In this paper we investigate the benefits of exploiting a semantic-based expansion of queries and documents that combines the use of Linked Open Data (LOD) and Knowledge Extraction (KE) techniques. KE techniques, implemented in state-of-the-art approaches such as FRED [1], NewsReader [2] and PIKES [3], exploit Natural Language Processing (NLP) methods to extract semantic content from textual resources, such as queries and documents. Extracted content is expressed (and disambiguated) using identifiers and vocabulary terms from well-established LOD resources (e.g., DBpedia [4], YAGO [5]), thus connecting to a growing body of LOD background knowledge from which related assertional and terminological knowledge can be injected in the IR task.

[*] Corresponding author. E-mail: rospocher@fbk.eu

1570-0844/0-1900/$35.00 © 0 – IOS Press and the authors. All rights reserved

This way, queries and documents can be expanded with additional *semantic terms* not explicitly mentioned in them. In particular, the semantic-based expansion that we consider includes terms from the following *semantic layers*:

1. *entities*, e.g., term dbpedia:Carl_Friedrich_Gauss extracted from mention "Gauss" in query "astronomers influenced by Gauss";
2. *types* of entities, either explicitly mentioned, such as yago:Astronomer109818343 from "astronomers", or indirectly obtained from external resources for mentioned entities, such as yago:GermanMathematicians obtained from mention "Gauss" (via dbpedia:Carl_Friedrich_Gauss);
3. *temporal information*, either explicitly mentioned in the text or indirectly obtained from external resources for mentioned entities, e.g., via DBpedia properties such as dbo:dateOfBirth (1777) and dbo:dateOfDeath (1855) for mentioned entity dbpedia:Carl_Friedrich_Gauss; and,
4. *semantic frames* and *frame roles*, such as term ⟨framebase:Subjective_influence, dbpedia:Carl_Friedrich_Gauss⟩ derived from "influenced by Gauss".

We then match query and documents considering both their textual and semantic content, according to a simple retrieval model based on the Vector Space Model (VSM) [6]. This way, we can match documents mentioning someone who is an astronomer (i.e., entities of type yago:Astronomer109818343) even if "astronomers", or one of its textual term-based variants, is not explicitly written in the document. Similarly, we can exploit the entities and the temporal content to better weigh the relevance of documents mentioning dbpedia:Carl_Friedrich_Gauss vs. dbpedia:GAUSS_(software), as well as to differently rank documents about Middle Age and 17th/18th centuries astronomers.

We implemented the approach in a system, called KE4IR (read: *kee-fer*), that exploits PIKES for the KE analysis of queries and documents, and Apache Lucene[1] for indexing the document collection and computing the relevance between queries and documents. A preliminary assessment of the approach was conducted in [7], where KE4IR was evaluated on a recently released, small-size dataset (WES2015) [8]. Those preliminary results gave hints that enriching textual information with semantic content outperforms retrieval performances over using textual data only. In this paper we build on those results, assessing KE4IR performances on several additional large-scale datasets (TREC Ad-hoc, TREC WT10g, the F&al. dataset described in [9]).

These new evaluations allow to:

- strengthen many of the findings in [7], confirming that the addition of semantic content enables constantly outperforming the retrieval performances obtained using textual data only;
- show that KE4IR performs better than a reference semantic-based IR approach [9], that builds only on ontology terminological knowledge (e.g., types, subtypes);
- give evidence that performing KE, enriching with LOD content, indexing, and searching collections up to millions of documents with KE4IR is feasible.

As for [7], we release all the synthetic evaluation results, the code (including evaluation scripts), and the auxiliary data we used (TREC datasets excluded due to copyright restrictions) on our website,[2] to allow replicating and extending our work and experiments.

While other works (e.g. [7–9]) have given evidences that semantic technologies are capable to enhance (text-based) document retrieval, the evaluation conducted in this paper — on a total of 555 queries over more than 2.2 million documents from different collections — provides a solid, unprecedented assessment of the impact of semantic technologies for the document retrieval task.

The paper is structured as follows. In Section 2, we review the state of the art in IR and KE. Section 3 presents the KE4IR approach, detailing the semantic layers and the retrieval model used for combining semantic and textual information. In Section 4, we describe the actual implementation of KE4IR, while in Section 5, we report on the comprehensive assessment over several datasets of the effectiveness of adding semantic content for IR, discussing in details some outcomes and findings in Section 6. Section 7 concludes with some final remarks and future work directions.

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

[2] http://pikes.fbk.eu/ke4ir.html

2. State of the Art

Previous works have exploited some semantic information for IR. An early tentative in injecting domain knowledge information for improving the effectiveness of IR systems is presented in [10]. In this work, authors manually built a thesaurus supporting the expansion of terms contained in both documents and queries. Such a thesaurus models a set of relations between concepts including synonymy, hyponymy and instantiation, meronymy and similarity. An approach based on the same philosophy is presented in [11], where the authors propose an indexing technique where WordNet [12] synsets, extracted from each document word, are used in place of textual terms in the indexing task. An evolved version of such approach is described in [13] where each synset is weighted accordingly to its number of explicit and implicit occurrences.

In the last decade, semantic IR systems started to embed ontologies for addressing the task of retrieving domain-specific documents. An interesting review on IR techniques based on ontologies is presented in [14], while in [15] the author studies the application of ontologies to a large-scale IR system for Web usage. Two models for the exploitation of ontology-based knowledge bases are presented in [16, 17]. The aim of these models is to improve search over large document repositories. Both models include an ontology-based scheme for the annotation of documents, and a retrieval model based on an adaptation of the classic Vector Space Model (VSM) [6]. A general IR system aimed at facilitating domain specific search is illustrated in [18]. The system uses fuzzy ontologies and is based on the notion of “information granulation”, a computational model aiming at estimating the granularity of documents. The presented experiments confirm that the proposed system outperforms a vector space based IR system for domain specific search. A further work exploring the use of semantic similarity measures for ontology-based IR has been presented in [19]. The main difference between the discussed approach and traditional VSM extensions is that it relies on Yager’s aggregation operators — preference models that capture end user expectations — for performing a direct assessment of semantic similarity analysis. More in details, queries are expressed using concepts of an ontology applied to index documents. The system estimates the overall relevance of documents w.r.t. a given query, obtained by aggregating, via Yager’s operators, the *partial similarity measurements* between each ontology concept of the query and those indexing the document (i.e., each document is scored w.r.t. every single concept of the query). Then, as a result of the proposed aggregation method, the most relevant documents (indexed with the query concepts) are ranked higher than least relevant ones (indexed with more general or more specific concepts than query’s ones). Finally, in [20] an analysis of the usefulness of ontologies for the retrieval task is discussed.

More recently, approaches combining many different semantic resources for retrieving documents have been proposed. In [9], the authors describe an ontology-enhanced IR platform where a repository of domain-specific ontologies is exploited for addressing the challenges of IR in the massive and heterogeneous Web environment. Given a query, this is annotated with concepts extracted from ontologies modeling the domains that the query belongs to. Documents of the collection used for evaluating the approach are annotated by using the Wraetlic NLP Suite[3] to enrich them with representative concepts that ease the retrieval process. While on the one hand the presented approach represents a full-fledged solution for semantic IR, on the other hand it suffers from requiring specific ontologies for performing the query annotation task. This drawback is avoided by approaches leveraging DBpedia [4] and other established LOD datasets as general-purpose sources of knowledge, like the retrieval models recently presented in [8, 21] and assessed on one of the evaluation datasets (WES2015) considered for KE4IR in this paper (see Section 5.1.4).

A further problem in IR is the ranking of retrieved results. Users typically make short queries and tend to consider only the first ten to twenty results [22]. In [23], a novel approach for determining relevance in ontology-based IR is presented, different from VSM. When IR approaches are applied in a real-world environment, the computational time needed to evaluate the match between documents and the submitted query has to be considered too. Systems using VSM have typically higher efficiency with respect to systems that adopt more complex models to account for semantic information. For instance, the work in [24] implements a non-vectorial data structure with high computational times for both indexing and retrieving documents.

In [7], we firstly presented KE4IR, an approach for document retrieval that exploits KE techniques and LOD resources. The work stemmed from the recent

[3] http://alfonseca.org/eng/research/wraetlic.html

advances in KE, resulting in several approaches and tools capable of performing comprehensive analyses of text to extract quality knowledge. Among them: FRED [1], a tool that extracts Discourse Representation Structures, mapping them to linguistic frames in VerbNet[4] and FrameNet,[5] which in turn are transformed in RDF/OWL via Ontology Design Patterns;[6] NewsReader [2], a comprehensive processing pipeline that extracts and corefers events and entities from large (cross-lingual) news corpora; and, PIKES[7] [3, 25], an open-source frame-based KE framework that combines the processing of various NLP tools to distill knowledge from text, aligning it to LOD resources such as DBpedia and FrameBase[8] [26] (a broad-coverage SW-ready inventory of frames based on FrameNet).

In particular, KE4IR builds on PIKES for analyzing queries and documents, and linking them to external knowledge resources from which related semantic information (i.e., background knowledge) can be imported for use in the retrieval task. To the best of our knowledge, KE4IR is the first approach that applies comprehensive KE techniques on documents and queries to improve document retrieval performances. Besides ontological types, Wordnet synsets, and named entities — the kind of semantic content previously considered by other state-of-the-art IR approaches — KE4IR leverages additional knowledge, such as frames and time information, made available by the KE techniques exploited. To accommodate and effectively exploit this additional content, KE4IR relies on a specifically developed adaptation of VSM, that accounts (possibly) for multiple semantic content available on a single textual term, and the fact that the same semantic content may originate from different terms in the query/document. A detailed description of the latest development of KE4IR, as used in this paper, is provided in Section 4.

Previous works (e.g. [7–9]) have shown evidences that semantic technologies are capable to enhance (text-based) document retrieval. However:

- [7, 8] report evaluation results only on a single, small dataset (WES2015, with 35 queries and 331 documents) compared to traditional document retrieval datasets which consists of collections of millions of documents;
- [9] was evaluated only on a single dataset (F&al., with 20 queries and 1.6 million documents) and a small number of queries;
- only [7] reports details on the statistical significance of the achieved results.

Compared to these works, the evaluation presented in this paper is conducted on multiple datasets (including general — i.e. not specifically devised for semantic technologies — state-of-the-art document retrieval datasets) with 4 document collections and 12 query sets, for a total of 555 queries over more than 2.2M documents, an unprecedented evaluation setting for semantic technologies in IR. Given the size and the variety of the considered document collections and query sets, we believe the work presented in this paper provides a solid assessment of the impact of semantic technologies for the document retrieval task.

It is worth noting that, beside the well-known document retrieval task, knowledge representation features have been used also for improving the effectiveness of systems for question answering [27–30]. This task consists in answering a user's unstructured query with a structured response taken from a knowledge base. As such, this task is substantially different from document retrieval one, investigated in this work.

3. Approach

Standard IR systems treat documents and queries as bags of *textual terms* (i.e., stemmed tokens). In KE4IR we consider additional *semantic terms* coming from semantic annotation layers produced using NLP-based KE techniques and LOD background knowledge (Section 3.1), and we propose a retrieval model using this additional semantic information to find and rank the documents matching a query (Section 3.2).

3.1. Semantic Layers

We consider four semantic layers — URI, TYPE, TIME, FRAME — that complement the TEXTUAL layer with semantic terms.[9] These terms are extracted from the RDF knowledge graph obtained from a text using KE techniques. This graph contains a structured rep-

[4] http://verbs.colorado.edu/

[5] http://framenet.icsi.berkeley.edu/

[6] http://ontologydesignpatterns.org/

[7] http://pikes.fbk.eu/

[8] http://framebase.org/

[9] Additional layers (e.g., location) are conceivable and may be worth investigating if they can provide enough semantic terms.

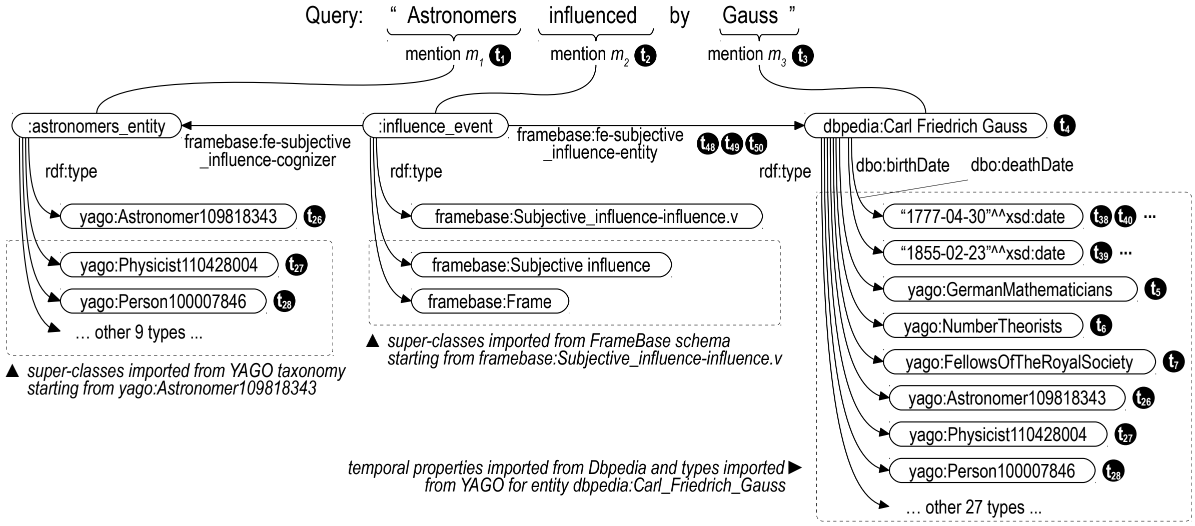

Query: " Astronomers influenced by Gauss "
mention m_1 t1
mention m_2 t2
mention m_3 t3
:astronomers_entity
framebase:fe-subjective_influence-cognizer
:influence_event
framebase:fe-subjective_influence-entity
t48 t49 t50
dbpedia:Carl Friedrich Gauss t4
rdf:type
rdf:type
rdf:type
dbo:birthDate
dbo:deathDate
yago:Astronomer109818343 t26
yago:Physicist110428004 t27
yago:Person100007846 t28
... other 9 types ...
▲ super-classes imported from YAGO taxonomy starting from yago:Astronomer109818343
framebase:Subjective_influence-influence.v
framebase:Subjective influence
framebase:Frame
▲ super-classes imported from FrameBase schema starting from framebase:Subjective_influence-influence.v
"1777-04-30"^^xsd:date t38 t40 ···
"1855-02-23"^^xsd:date t39 ···
yago:GermanMathematicians t5
yago:NumberTheorists t6
yago:FellowsOfTheRoyalSociety t7
yago:Astronomer109818343 t26
yago:Physicist110428004 t27
yago:Person100007846 t28
... other 27 types ...
temporal properties imported from Dbpedia and types imported ▶ from YAGO for entity dbpedia:Carl_Friedrich_Gauss

Fig. 1. RDF knowledge graph and terms extracted from "astronomers influenced by Gauss". The top of the graph (:astronomers_entity, :influence_event, dbpedia:Carl_Friedrich_Gauss, their links and most-specific types) comes from KE, while the rest comes from background knowledge resources: DBpedia, YAGO, FrameBase.

Table 1

Terms extracted from the example query "astronomers influenced by Gauss", with mentions m_1 = "astronomers", m_2 = "influenced", m_3 = "Gauss", $\mathrm{w}(l) = 0.5$ for the TEXTUAL layer, $\mathrm{w}(l) = 0.125$ for each semantic layer; idf values computed on the WES2015 dataset (Section 5.1).

	Layer l	**Term t_i**	$M(t_i, q)$	$\mathrm{tf}_q(t_i, q)$	$\mathrm{idf}(t_i, q)$	$\mathrm{w}(l)$	q_i
t_1	TEXTUAL	astronom	m_1	1.0	2.018	0.5	1.009
t_2	TEXTUAL	influenc	m_2	1.0	3.404	0.5	1.702
t_3	TEXTUAL	gauss	m_3	1.0	1.568	0.5	0.784
t_4	URI	dbpedia:Carl_Friedrich_Gauss	m_3	1.0	3.404	0.125	0.426
t_5	TYPE	yago:GermanMathematicians	m_3	0.030	2.624	0.125	0.010
t_6	TYPE	yago:NumberTheorists	m_3	0.030	2.583	0.125	0.010
t_7	TYPE	yago:FellowsOfTheRoyalSociety	m_3	0.030	1.057	0.125	0.004
…	TYPE	… other 18 terms …	m_3	0.030	…	0.125	…
t_{26}	TYPE	yago:Astronomer109818343	m_1, m_3	0.114	1.432	0.125	0.020
t_{27}	TYPE	yago:Physicist110428004	m_1, m_3	0.114	0.958	0.125	0.014
t_{28}	TYPE	yago:Person100007846	m_1, m_3	0.114	0.003	0.125	~0
…	TYPE	… other 9 terms …	m_1, m_3	0.114	…	0.125	…
t_{38}	TIME	day:1777-04-30	m_3	0.1	3.404	0.125	0.043
t_{39}	TIME	day:1855-02-23	m_3	0.1	3.404	0.125	0.043
t_{40}	TIME	century:17	m_3	0.1	0.196	0.125	0.002
…	TIME	… other 7 terms	m_3	0.1	…	0.125	…
t_{48}	FRAME	⟨Subjective_influence-influence.v, dbpedia:Carl_Friedrich_Gauss⟩	m_2	0.333	5.802	0.125	0.242
t_{49}	FRAME	⟨Subjective_influence, dbpedia:Carl_Friedrich_Gauss⟩	m_2	0.333	5.802	0.125	0.242
t_{50}	FRAME	⟨Frame, dbpedia:Carl_Friedrich_Gauss⟩	m_2	0.333	3.499	0.125	0.146

resentation of the entities, events, and relations mentioned in the text, each one linked to the specific snippets of text, called *mentions*, that denote that element, as shown in Figure 1 for the example text of Section 1: "astronomers influenced by Gauss". From each mention, a set of semantic terms is extracted by considering the elements of the knowledge graph rooted at that mention, as explained later for each layer and as exemplified in Table 1 (first four columns) for the considered example. A mention may express multiple semantic terms (differently from the textual case) and a semantic term may originate from multiple mentions, whose number can be used to quantify the relevance of the term for a document or query.

URI layer This layer consists of the URIs of entities mentioned in the text, disambiguated against external knowledge bases such as DBpedia (i.e., at most one entity in the background knowledge is linked to an entity mention). Disambiguated URIs result from two NLP/KE tasks:[10] Named Entity Recognition and Classification (NERC), which identifies proper names of certain entity categories (e.g., persons, organizations, locations) in a text, and Entity Linking (EL), which disambiguates those names against the individuals of a knowledge base. The Coreference Resolution NLP task can be also exploited to "propagate" a disambiguated URI from a mention to another coreferring mention in the text, to better count the number of entity mentions for each URI. In the example of Figure 1, the URI term dbpedia:Carl_Friedrich_Gauss (t_4) is extracted from the corresponding DBpedia entity mentioned as "Gauss" in the text.

TYPE layer The terms of this layer are the URIs of the ontological types (and super-types) associated to entities of any kind mentioned in the text. For disambiguated named entities with a URI (from NERC and EL), associated types are obtained from LOD background knowledge describing those entities, like TYPE term yago:NumberTheorists (t_6) obtained from the DBpedia description of entity dbpedia:Carl_Friedrich_Gauss, in Figure 1. For other entities and common nouns, disambiguation against WordNet through Word Sense Disambiguation (WSD) returns synsets that can be mapped to ontological types via existing mappings, like TYPE term yago:Astronomer109818343 (t_{26}) obtained by disambiguating word "Astronomers" to WordNet 3.0 synset n-09818343. An ontology particularly suited to both extraction techniques is the YAGO taxonomy [5], as its types are associated to WordNet synsets as well as DBpedia entities.

TIME layer The terms of this layer are the temporal values related to the text, either because explicitly mentioned in a time expression (e.g., the text "eighteenth century") recognized through the Temporal Expression Recognition and Normalization (TERN) NLP task, or because associated to a disambiguated entity via some property in the background knowledge, such as the birth date 1777-04-30 associated to dbpedia:Carl_Friedrich_Gauss in the example of Figure 1 and Table 1. To support both precise and fuzzy temporal matching of queries and documents, each temporal value is represented with (max) five TIME terms at different granularity levels — day, month, year, decade, and century — as, e.g., value 1777-04-30 mapped to terms day:1777-04-30 (t_{38}), month:1777-04, year:1777, decade:177, century:17 (t_{40}).

FRAME layer A semantic frame is a star-shaped structure representing an event or n-ary relation, which has a frame type and zero or more participants each playing a specific semantic role in the context of the frame. An example of frame is :influence_event in Figure 1, having type framebase:frame-Subjective_influence (among others) and participants dbpedia:Carl_Friedrich_Gauss (a disambiguated entity) and :astronomers_entity (a non-disambiguated entity). Semantic frames can be extracted using NLP tools for Semantic Role Labeling (SRL), which are based on predicate models that define frame types and roles, such as FrameNet. The outputs of these tools are then mapped to an ontological representation using an RDF/OWL frame-based ontology aligned to the predicate model, such as FrameBase [26]. Semantic frames provide relational information that can be leveraged to match queries and documents more precisely. To this end, in KE4IR we map each $\langle$frame type, participant$\rangle$ pair whose participant is a disambiguated entity, such as pair $\langle$framebase:frame-Subjective_influence, dbpedia:Carl_Friedrich_Gauss$\rangle$ in Figure 1, to a FRAME term (t_{49}), including also the terms obtainable by considering the frame super-types in the ontology (the use of non-disambiguated participant entities leads to worse retrieval results).

We remark that, both for documents and queries, only their textual content is used as context for extracting semantic terms via NLP-based KE techniques and LOD background knowledge. While the textual context available for documents is generally long enough, the context for queries is much more limited given their short nature, although it is still present (few user keywords in the query may still allow for some effective disambiguation, as one may experience in the everyday use of search engines). In this work we do not consider additional context for queries beyond their textual content, coherently with the benchmarks used for evaluating our approach (e.g., TREC — see Section 5 for details), where only the query text is considered. In real scenarios, search engines may have access to past user queries and click stream, which provide additional context for query disambiguation.

[10] We briefly mention in this section the main NLP/KE tasks involved in the extraction of semantic terms. Some of these tasks typically build on additional NLP analyses, such as Tokenization, Part-of-Speech tagging, Constituency Parsing and Dependency Parsing.

3.2. Retrieval Model

The KE4IR retrieval model is inspired by the Vector Space Model (VSM). Given a document collection D, each document $d \in D$ (resp. query q) is represented with a vector $\mathbf{d} = (d_1 \dots d_n)$ ($\mathbf{q} = (q_1 \dots q_n)$) where each element d_i (q_i) is the weight corresponding to term t_i, while n is the number of distinct terms in the collection D. Differently from text-only approaches, the terms of our model come from multiple layers, both textual and semantic, and each document (query) vector can be seen as the concatenation of smaller, layer-specific vectors [31]. Given a term t, we denote the layer it belongs to with $\mathrm{l}(t) \in L = \{\mathrm{TEXTUAL}, \mathrm{URI}, \mathrm{TYPE}, \mathrm{TIME}, \mathrm{FRAME}\}$.

The goal of the retrieval model is to compute a similarity score $sim(d,q)$ between each document $d \in D$ and query q. The documents matching query q are the ones with $sim(d,q) > 0$, and they are ranked based on decreasing similarity values. To derive $sim(d,q)$, we start from the definition of the cosine similarity used in VSM:

$$sim_{\mathrm{VSM}}(d,q) = \frac{\mathbf{d} \cdot \mathbf{q}}{\|\mathbf{d}\|_2 \cdot \|\mathbf{q}\|_2} = \frac{\sum_{i=1}^{n} d_i \cdot q_i}{\sqrt{\sum_{i=1}^{n} d_i^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}} \quad (1)$$

and we remove the normalizations by the Euclidean norms $\|\mathbf{d}\|_2$ and $\|\mathbf{q}\|_2$, obtaining:

$$sim(d,q) = \mathbf{d} \cdot \mathbf{q} = \sum_{i=1}^{n} d_i \cdot q_i \quad (2)$$

Normalizing by $\|\mathbf{q}\|_2$ serves to compare the scores of *different* queries and does not affect the ranking, thus we drop it for simplicity. Normalizing by $\|\mathbf{d}\|_2$ makes the similarity score obtained by matching m query terms in a small document higher than the score obtained by matching the same m query terms in a longer document. This normalization is known to be problematic in some document collections (it is defined differently and optionally disabled in production systems such as Lucene and derivatives) and we consider it inappropriate in our scenario, where the document vector is expanded with large amounts of semantic terms whose number depends not just on the document length (as for textual terms) but also on the richness of the entity descriptions those semantic terms are derived from, both in the RDF knowledge graph extracted from text and the background knowledge.

To assign the weights of document and query vector elements, we adopt the usual product of Term Frequency (tf) and Inverse Document Frequency (idf):

$$d_i = \mathrm{tf}_d(t_i, d) \cdot \mathrm{idf}(t_i, D) \quad (3)$$

$$q_i = \mathrm{tf}_q(t_i, q) \cdot \mathrm{idf}(t_i, D) \cdot \mathrm{w}(\mathrm{l}(t_i)) \quad (4)$$

The values of tf are computed in different ways for documents (tf_d) and queries (tf_q), while the weights $\mathrm{w}(\mathrm{l}(t_i))$, with $\mathrm{w}(l) \geqslant 0$ for all $l \in L$ and $\sum_{l \in L} \mathrm{w}(l) = 1$, are "hyper-parameters" of our approach that permit balancing the contribution of different layers to the final similarity score.[11] Given the form of Equation 2, it suffices to apply $\mathrm{w}(\mathrm{l}(t_i))$ only to one of $\mathbf{d}$ and $\mathbf{q}$; we chose $\mathbf{q}$ to allow selecting weights on a per-query basis.[12] Table 1 (last four columns) reports the tf_q, idf, w, and q_i values for the terms of the example query "astronomers influenced by Gauss".

Several schemes for computing tf and idf have been developed in the literature. Given $f(t,x)$ and $f'(t,x)$ two ways of measuring the frequency of a term t in a text (document or query) x, we adopt the scheme:[13]

$$\mathrm{tf}_d(t,d) = 1 + \log(f(t,d)) \quad (5)$$

$$\mathrm{tf}_q(t,q) = f'(t,q) \quad (6)$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D | f(t,d) > 0\}|} \quad (7)$$

where $\mathrm{tf}_d(t,d)$ and $\mathrm{idf}(t,D)$ are set to 0 if the referred term t does not appear respectively in document d or corpus D (as logarithm and division are undefined).

The *raw frequency* $f(t,x)$ is defined as usual as the number of times term t occurs in text x. To account for semantic terms, we denote with $M(t,x)$ the set of mentions in text x from where term t was extracted, valid also for textual terms whose mentions are simply their occurrences in the text, and define $f(t,x) = |M(t,x)|$. The *normalized frequency* $f'(t,x)$, instead, is newly introduced to account for the fact that in a semantic layer multiple terms can be extracted from a single mention,

[11] $\mathrm{w}(l) = 0$ indicates that layer l has no effect at all on similarity, while $\mathrm{w}(l) = 1$ indicates that l is the only layer affecting similarity as other layers l' must have $\mathrm{w}(l') = 0$. Moving from 0 to 1 the layer "importance" increases, although not necessarily linearly.

[12] Alternatively, $\mathrm{w}(\mathrm{l}(t_i))$ may be introduced with the same effects in Equation 2, i.e., $sim(d,q) = \sum_{i=1}^{n} d_i \cdot q_i \cdot \mathrm{w}(\mathrm{l}(t_i))$.

[13] Our scheme can be classified as ltn.ntn using the SMART notation used in the literature; see http://bit.ly/weighting_schemes [32].

differently from the textual case. It is defined as:

$$f'(t,x) = \sum_{m \in M(t,x)} \frac{1}{|T(m, \mathrm{l}(t))|} \qquad (8)$$

where $T(m,l)$ denotes the set of terms of layer l extracted from mention m. Since $|T(m, \textsc{textual})| = 1$ for any mention, $f(t,x) = f'(t,x)$ for any textual term. Note that $f(t,x)$ or $f'(t,x)$ can be indifferently used in Equation 7.

The formulation of $f'(t,x)$ and its use in Equation 6 aim at giving each mention the same importance when matching a query against a document collection. To explain, consider a query with two mentions m_1 and m_2, from which respectively n_1 and n_2 disjoint terms of a certain semantic layer (e.g., TYPE) were extracted, $n_1 > n_2$; also assume that these terms have equal idf and tf_d values in the document collection. If we give these terms equal tf_q values (e.g., $tf_q(t,q) = f(t,q) = 1$ as their raw frequency), then a document matching the n_1 terms of m_1 (and nothing else) will be scored and ranked higher than a document matching the n_2 terms of m_2 (and nothing else). However, the fact that $n_1 > n_2$ does not reflect a preference of m_1 by the user; rather, it may reflect the fact that m_1 is described more richly (and thus, with more terms) than m_2 in the background knowledge. Our definition of normalized frequency corrects for this bias by assigning each query mention a total weight of 1 for each semantic layer, which is equally distributed among the terms extracted from the mention for that layer (e.g., weight $1/n_1$ for terms of m_1, $1/n_2$ for terms of m_2).

Similarly, the use of $f'(t,x)$ in place of $f(t,x)$ in Equation 5 would be inappropriate. Consider a query whose vector has a single TYPE term t (similar considerations apply to other semantic layers). Everything else being equal (e.g., idf values), two documents mentioning two entities of type t the same number of times should receive the same score. While this happens when using $f(t,x)$ for $tf_d(t,d)$, with $f'(t,x)$ the document mentioning the entity with fewest TYPE terms (beyond t) would be scored higher, although this clearly does not reflect a user preference.

4. Implementation

We built an evaluation infrastructure including an implementation of the KE4IR approach presented in Section 3. The infrastructure allows the batch application and assessment of KE4IR on arbitrary documents and queries with their gold relevance judgments. All the source code, binaries, and necessary data are available on KE4IR website.[14]

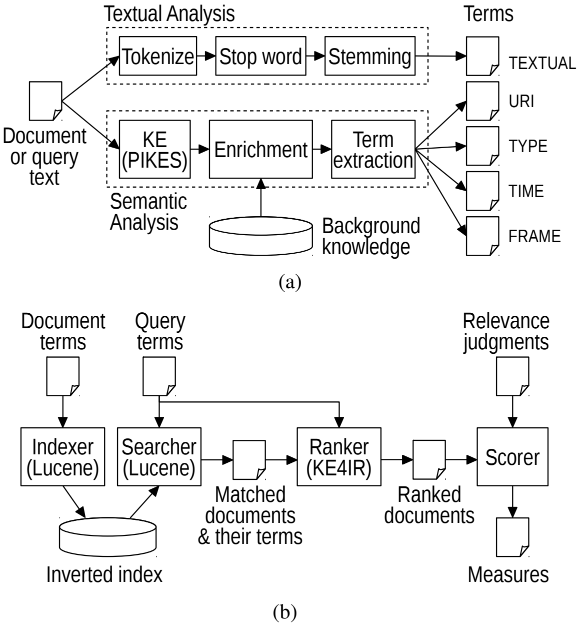


Fig. 2. KE4IR implementation: (a) term extraction for documents and queries; (b) query execution and evaluation against gold relevance judgments.

Figure 2a shows the pipeline used to map the text of a document or query to a set of terms, combining both textual and semantic analysis. Textual analysis aims at extracting textual terms through the usual combination of text tokenization, stop word filtering, and stemming techniques, implemented using standard components from Apache Lucene. Semantic analysis, instead, aims at extracting semantic terms of the four layers considered in KE4IR. It uses a KE tool (PIKES) to transform the input text into an RDF knowledge graph whose nodes (RDF URIs and literals) are entities mentioned in the text, entity types, and property values, and whose edges (RDF triples) describe entities and their relations. Well-known nodes in the graph (DBpedia entities, YAGO and FrameBase concepts) are enriched with additional triples about them from a persistent key-value store populated with LOD background knowledge (YAGO and FrameBase schemas, DBpedia mapping-based properties with xsd:date, xsd:dateTime, xsd:gYear, and xsd:gYearMonth objects, to provide for additional tem-

[14] http://pikes.fbk.eu/ke4ir.html

poral information). As both ABox and TBox triples are inserted, RDFS reasoning is applied to materialize inferable rdf:type triples that affect the extraction of TYPE and FRAME terms.[15] The resulting graph is finally queried to extract semantic terms according to their definitions as of Section 3.1.

Figure 2b shows the pipeline that accepts extracted document and query terms, executes queries according to the KE4IR model, and computes evaluation metrics against gold relevance judgments. To efficiently find the documents matching a query in large collections, the pipeline employs a Lucene inverted index populated with the term vectors of the documents in the collection, including their raw frequencies. When executing a query q, its terms are OR-ed together to form a Lucene boolean query that is evaluated on the index and returns the list of matching documents d containing at least one query term, so that $sim(d, q) > 0$. A ranker component external to Lucene (for ease of testing) implements the KE4IR model of Section 3.2 and is responsible for ranking the matched documents, based on their term vectors extracted from the index, the term vector of the query, and some index statistics (number of documents and document frequencies) required to compute idf values. A scorer component compares the resulting ranking with the gold relevance judgments, computing a comprehensive set of evaluation measures that are averaged along queries.

The most complex and computationally expensive task in KE4IR implementation is KE. We use PIKES, a frame-based KE framework providing state-of-the-art performances via a 2-phase approach. In the first *linguistic feature extraction* phase, an RDF graph of mentions is obtained by running and combining the outputs of several state-of-the-art NLP tools, including Stanford CoreNLP[16] (tokenization, lemmatization, POS-tagging, TERN, NERC, coreference resolution, parsing), DBpedia Spotlight[17] (EL), UKB[18] (WSD), Semafor[19] and Mate-tools[20] (SRL). In the second *knowledge distillation* phase, the mention graph is transformed into an RDF knowledge graph through the evaluation of mapping rules, using the RDFpro[21] [33] tool for RDF processing that we also use for RDFS reasoning in Figure 2a. Using multiple instances of PIKES on a server with 12 cores (24 threads) and 192 GB RAM, we obtained a throughput of ~700K tokens/h (~30K tokens/h per core), corresponding to ~3180 documents/h for the average document length of 220 tokens observed in our experiments. Processing time is almost totally spent in the NLP analysis of texts. By mapping KE4IR semantic layers to the required NLP tasks, their impact on the whole processing time results to be: 3.5% URI, 16.3% TYPE, 2.9% TIME, 77.3% FRAME.

Note that the current version of PIKES works only on English texts, and thus the current KE4IR implementation can be applied out-of-the-box only on English document collections. However, the adaptation to different languages requires changes only on the term extraction pipeline in Figure 2a (on the textual and semantic analysis steps, and possible the enrichment/background knowledge step) and no changes are needed on the query execution pipeline in Figure 2b.

Regarding entity linking, in line with what noted in Section 3.1, a mention is linked to at most one entity in DBpedia by DBpedia Spotlight, even if multiple candidate entities might be proposed for it (e.g., dbpedia:Carl_Friedrich_Gauss, dbpedia:Gauss_(unit) for entity mention "Gauss");[22] if multiple candidate entities are equally likely (i.e., they get the same confidence score), the most popular is chosen lacking additional context for disambiguation.

5. Evaluation

In this section, we empirically assess whether the enriching of document and query term vectors with semantic terms significantly affects IR performances. More precisely, we investigate the following research question:

[15] Specifically, we only need to materialize implicit rdf:type triples based on rdfs:domain, rdfs:range, rdfs:subClassOf, and rdfs:subPropertyOf TBox axioms in background knowledge. This inference is inexpensive compared to NLP analysis, and most of it can be done in a pre-processing step that computes the closure of background knowledge, simplifying inference at document/query analysis time.

[16] http://nlp.stanford.edu/software/corenlp.shtml

[17] http://spotlight.dbpedia.org/

[18] http://ixa2.si.ehu.es/ukb/

[19] http://www.cs.cmu.edu/~ark/SEMAFOR/

[20] http://code.google.com/p/mate-tools/

[21] http://rdfpro.fbk.eu/

[22] The choice to consider only the best matching entity (if any) and not all the candidate entities matching a mention is a result of our focus on KE as a way to obtain semantic terms for a query or document text. In our approach, the text is first interpreted via KE to obtain a corresponding knowledge graph, out of which semantic terms are extracted. Since in the knowledge graph a mention denotes exactly one entity, there is no natural way in this approach to consider also the other non-best candidate entities that were produced by DBpedia Spotlight.

Table 2

Main characteristics of the document collections used in the evaluations. TREC Disk 4&5(-) identifies the TREC Disk 4&5 document collection from which congressional records are removed.

Collection	**Document Type**	**Size**	**(% of) Documents having Layer**					**Terms per Layer (avg. on docs)**				
			TEXTUAL	URI	TYPE	TIME	FRAME	TEXTUAL	URI	TYPE	TIME	FRAME
WES2015	blog posts	331	100.00	100.00	100.00	99.70	99.70	282.13	22.11	434.29	75.39	176.53
TREC Disk 4&5	news, notices, congress. records	555949	100.00	98.91	100.00	94.13	95.85	206.39	12.53	284.53	41.39	77.12
TREC Disk 4&5(-)	news, notices	528027	100.00	98.86	100.00	93.85	95.78	202.55	12.05	280.77	40.24	74.79
TREC WT10g	crawled web pages	1687241	100.00	97.29	99.15	88.24	85.65	225.86	25.05	274.36	43.49	67.66

Table 3

Main characteristics of the query sets used in the evaluations.

Query Set	**Size**	**(% of) Queries having Layer**					**Terms per Layer (avg. on queries)**					**Collection**	**Relevance Judgments**	**Evaluation Section**
		TEXTUAL	URI	TYPE	TIME	FRAME	TEXTUAL	URI	TYPE	TIME	FRAME			
WES2015	35	100.00	71.43	94.29	31.43	31.43	3.57	1.16	26.18	11.55	4.82	WES2015	5-value (0-4)	Section 5.1
F&al.2011	20	100.00	35.00	95.00	10.00	15.00	5.15	1.00	20.37	5.00	8.00	WT10g	binary (0-1)	Section 5.2
TREC 6 title	50	100.00	18.00	98.00	6.00	8.00	2.64	1.22	15.53	11.67	5.00	TREC Disk 4&5	binary (0-1)	Section 5.3
TREC 7 title	50	100.00	20.00	92.00	4.00	6.00	2.56	1.00	15.41	8.00	4.00	TREC Disk 4&5(-)	binary (0-1)	
TREC 8 title	50	100.00	24.00	98.00	16.00	4.00	2.46	1.17	15.63	11.25	4.50	TREC Disk 4&5(-)	binary (0-1)	
TREC 9 title	50	100.00	54.00	86.00	20.00	16.00	2.92	1.11	19.51	9.10	4.13	WT10g	binary (0-1)	
TREC 2001 title	50	100.00	28.00	92.00	18.00	16.00	3.80	1.07	19.63	7.11	4.00	WT10g	3-value (0-2)	
TREC 6 desc	50	100.00	38.00	100.00	16.00	30.00	13.78	1.26	36.76	10.88	7.27	TREC Disk 4&5	binary (0-1)	
TREC 7 desc	50	100.00	24.00	100.00	6.00	16.00	9.92	1.33	28.28	12.67	6.38	TREC Disk 4&5(-)	binary (0-1)	
TREC 8 desc	50	100.00	32.00	100.00	24.00	20.00	9.24	1.69	28.76	11.33	5.30	TREC Disk 4&5(-)	binary (0-1)	
TREC 9 desc	50	100.00	60.00	100.00	38.00	40.00	9.90	1.37	35.98	9.00	5.65	WT10g	binary (0-1)	
TREC 2001 desc	50	100.00	32.00	100.00	20.00	28.00	7.70	1.13	25.54	6.70	5.57	WT10g	3-value (0-2)	

RQ Does document and query enrichment with semantic terms enable to significantly outperform document retrieval when only raw textual information is used?

To address this research question, we conducted three evaluations with different datasets. Each evaluation consisted in performing a set of queries over a document collection for which the list of gold relevance judgments is available, and comparing the retrieved documents with such judgments. A summary of the main characteristics of the document collections and query sets used in each evaluation is reported in Tables 2 and 3.

For each dataset, we report KE4IR performances, comparing them with the ones of the *textual baseline*, obtained by indexing the raw text with Lucene tuned with the scoring formula of Section 4. In our experiments, this tuning provides the same performances of a standard Lucene configuration, and allows properly assessing the impact of semantic layers by excluding any interference related to slight differences in the scoring formula.

To assess the performances of KE4IR and the textual baseline we adopted the following measures:

- precision values after the first (Prec@1), fifth (Prec@5), and tenth (Prec@10) document, respectively. The rationale behind this choice is the fact that the majority of search result click activity (89.8%) happens on the first page of search results [22] corresponding to a set varying from 10 to 20 documents.
- Mean Average Precision (MAP), computed on the entire rank and after the first ten documents retrieved (MAP@10). Validation on the MAP metric enables assessing the capability of a system of returning relevant documents, independently of the ranking.
- Normalized Discounted Cumulated Gain (NDCG) [34], computed both on the entire rank and after the first ten documents retrieved (NDCG@10). Validation on the NDCG metric is necessary in scenarios where multi-value relevance is used.

This choice of measures makes our evaluation protocol analogous to the one adopted in the TREC [35] evaluation campaigns, with the addition of the NDCG metric. All the evaluation materials are available on KE4IR website.

In all three evaluations, the versions and the configurations of the NLP tools used in KE4IR are the default ones adopted in PIKES (see PIKES website for more details), which may be different from the default versions and configurations of the same tools as available online (e.g., the DBpedia Spotlight demo service[23]). For instance, in case of DBpedia Spotlight, PIKES uses a configuration tuned to optimize recall.

5.1. *WES2015 Dataset*

In this section, we summarize the results of a first evaluation of KE4IR on a recently released dataset [8], here referenced as WES2015, specifically developed to assess semantic search effectiveness. This evaluation (excluding the tuning of weights via brute force and RankLib at the end of Section 5.1.3, and the newly added comparison with other retrieval models in Section 5.1.4) was originally reported in [7] where additional details are provided.[24]

5.1.1. *Evaluation Set-up*

The peculiarity of the WES2015 collection and query set (see Tables 2 and 3 for their characteristics) is the underlying semantic purpose with which they were built. Indeed, the set of queries was selected by varying from queries very close to keyword-based search (e.g., query "Romanticism") to queries requiring semantic capabilities for retrieving relevant documents (e.g., "Aviation pioneers' publications"). We compare KE4IR against the textual baseline, initially using equal weights for textual and semantic information, i.e., $\mathrm{w}(\textsc{textual}) = \mathrm{w}(\textsc{semantics}) = 0.5$ with $\mathrm{w}(\textsc{semantics})$ divided equally among semantic layers, and then we study the impact on performances of using unequal layer weights. We conclude the section with a complementary comparison with the retrieval models proposed in [8, 21].

5.1.2. *Results*

Table 4 shows the comparison between the results achieved by KE4IR exploiting all the semantic layers with equal weights for textual and semantic information, and the results obtained by the textual baseline.

KE4IR outperforms the baseline for all the metrics, thus contributing to positively answer our research question. The highest improvements are registered on the MAP, MAP@10, and Prec@10 metrics that quantify the capability of KE4IR of producing an effective documents ranking when documents are considered either relevant or not relevant (i.e., binary relevance). On the other hand, the improvements on the NDCG and NDCG@10 metrics highlight that produced rankings are effective also from a qualitative point of view where the different degrees of relevance provided in WES2015 are taken into account. These improvements are statistically significant for MAP, MAP@10, NDCG, and NDCG@10 (significance threshold 0.05), based on the p-values computed with the one-tailed paired approximate randomization test [36].[25]

Table 5 reports all the dataset queries, the semantic layers for which terms were automatically identified via PIKES analyses, and the performances (on MAP and NDCG@10) of KE4IR and the textual baseline.[26]

The table shows that for NDCG@10 (resp. MAP) KE4IR outperform the textual baseline on 17 (resp. 19) queries out of 35, with improvements ranging from 0.002 (resp. 0.001) to 0.155 (resp. 0.283).

Queries q27 and q44 are examples where semantic information significantly boost performances. In q44, the correct link to dbpedia:Napoleon and the type and time information associated to that entity in DBpedia allow extracting URI, TYPE and TIME terms that greatly help ranking relevant documents higher. In q27, the major improvement derives from the extraction and matching of FRAME term ⟨framebase:frame-Destroying, dbpedia:Nazism⟩; while TIME information is also available (as dbpedia:Nazism is linked to category dbc:20th_century in DBpedia), our KE4IR implementation is not sophisticated enough to extract it.

Query q46 is an example where semantic information has no effects. This is because entities at different granularities are injected in the URI layers of query and documents. Specifically, the query is annotated with dbpedia:Nobel_Prize, while relevant documents have annotations like dbpedia:Nobel_Prize_in_X, where X is one of the disciplines for which Nobel Prizes are assigned. Unfortunately, these entities are not related in

[23] http://demo.dbpedia-spotlight.org/

[24] In [7], we also considered a baseline exploiting the Google custom search API for indexing pages containing our documents. However, this Google-based baseline was heavily outperformed by the textual baseline on the considered dataset, due to Google, being heavily tuned for precision, returning far less results than KE4IR and the textual baseline for the evaluation queries.

[25] We used the one-tailed approximate randomization test with alternative hypothesis that the mean of a given measure over queries for KE4IR is higher than the mean of the same measure for the textual baseline. The opposite alternative hypothesis, i.e., that KE4IR is worse than the baseline, is always rejected in our tests and we omit it. All the statistical significance results in this paper are confirmed also by the paired one-tailed t-test (p-values reported on KE4IR website), in line with [37] where both tests are found to produce similar results, and where approximate randomization is recommended.

[26] We selected only the MAP and the NDCG@10 metrics because those are the most indicative metrics for evaluating the performances of IR systems in general (MAP), and for deployment in a real-world environment (NDCG@10).

Table 4

Comparison of KE4IR against the Textual baseline (WES2015 query set and document collection).

Approach/System	**Prec@1**	**Prec@5**	**Prec@10**	**NDCG**	**NDCG@10**	**MAP**	**MAP@10**
Textual	0.943	0.669	0.453	0.832	0.782	0.733	0.681
KE4IR	**0.971**	**0.680**	**0.474**	**0.854**	**0.806**	**0.758**	**0.713**
KE4IR vs. Textual	3.03%	1.71%	4.55%	2.64%	2.99%	3.50%	4.74%
p-value (approx. random.)	0.500	0.251	0.055	**0.000**	**0.007**	**0.008**	**0.012**

Table 5

Queries, available semantic layers per query, and KE4IR vs textual baseline performances (NDGC@10 and MAP) on WES2015 [8].

Query		**Layers**				**NDCG@10**		**MAP**	
ID	**Text**	URI	TYPE	TIME	FRAME	**Textual**	**KE4IR**	**Textual**	**KE4IR**
q01	Fabrication of music instruments		X			0.589	0.589	0.478	0.478
q02	famous German poetry		X			0.634	0.634	0.724	**0.726**
q03	Romanticism	X	X			0.913	**0.915**	1.000	1.000
q04	University of Edinburgh research		X			0.895	0.895	0.962	0.962
q06	bridge construction		X			0.945	0.945	0.911	0.911
q07	Walk of Fame stars		X			0.405	0.405	0.314	**0.319**
q08	Scientists who worked on the atomic bomb		X			0.901	**0.950**	0.852	**0.864**
q09	Invention of the Internet	X	X			0.955	**0.984**	1.000	1.000
q10	early telecommunication methods		X			0.474	**0.56**7	0.441	**0.502**
q12	Who explored the South Pole	X	X		X	0.935	0.935	0.950	0.950
q13	famous members of the Royal Navy	X	X		X	0.913	**0.980**	0.927	0.927
q14	Nobel Prize winning inventions	X	X		X	0.497	**0.512**	0.622	**0.627**
q16	South America	X	X			0.755	**0.772**	0.417	**0.700**
q17	Edward Teller and Marie Curie	X	X	X		0.903	0.903	0.750	**0.775**
q18	Computing Language for the programming of Artificial Intelligence	X	X			0.981	**0.983**	0.937	**0.948**
q19	William Hearst movie	X	X	X		0.714	0.714	0.500	**0.513**
q22	How did Captain James Cook become an explorer	X	X	X	X	0.704	**0.717**	0.681	**0.701**
q23	How did Grace Hopper get famous	X	X	X	X	0.604	0.604	0.284	**0.299**
q24	Computers in Astronomy	X	X			0.369	0.369	**0.303**	0.284
q25	WWII aircraft	X	X			0.775	**0.930**	0.729	**0.854**
q26	Literary critics on Thomas Moore	X	X	X	X	1.000	1.000	1.000	1.000
q27	Nazis confiscate or destroy art and literature	X	X		X	0.631	**0.785**	0.522	**0.622**
q28	Modern Age in English Literature	X	X			**0.926**	0.809	**0.833**	0.738
q29	modern Physiology	X	X		X	0.965	**0.967**	0.833	0.833
q32	Roman Empire	X	X	X		1.000	1.000	1.000	1.000
q34	Scientists who have contributed to photosynthesis		X			1.000	1.000	1.000	1.000
q36	Aviation pioneers' publications		X			**0.914**	0.898	0.981	0.981
q37	Gutenberg Bible	X	X			0.864	0.864	0.667	**0.680**
q38	Religious beliefs of scientists and explorers		X			0.595	**0.623**	**0.559**	0.556
q40	Carl Friedrich Gauss influence on colleagues	X	X	X	X	0.827	**0.872**	0.898	**0.906**
q41	Personalities from Hannover	X	X	X		0.776	**0.897**	0.860	**1.000**
q42	Skinner's experiments with the operant conditioning chamber	X	X	X	X	0.699	**0.709**	0.454	**0.477**
q44	Napoleon's Russian Campaign	X	X	X		0.802	**0.953**	0.820	**0.967**
q45	Friends and enemies of Napoleon Bonaparte	X	X	X		**0.933**	0.931	0.931	**0.938**
q46	First woman who won a Nobel Prize	X	X		X	0.584	0.584	0.512	0.512
	Mean					0.782	**0.806**	0.733	**0.758**
	KE4IR vs. Textual					2.99%		3.50%	
	p-value (approx. random.)					**0.007**		**0.008**	

DBpedia (also in terms of types), thus it is not possible to expand the query in order to find matches with relevant documents.

Only in three cases for NDCG@10 (resp. MAP), KE4IR performs worse than the textual baseline. In particular, for query q28 worse performances are achieved, both on NDCG@10 and MAP, by using semantic information, due to Entity Linking errors. From the query, two URI terms (and related TYPE terms) are correctly extracted: dbpedia:Modern_history, with no matching documents, and dbpedia:English_literature, with 12 matches. Of these matches, 11 are incorrect and refer to irrelevant documents where dbpedia:English_literature is wrongly linked to mentions of other "English" things (e.g. "English scholar", "English society", "English medical herbs").

Complementary analyses on this dataset (e.g., impact of using different layer combinations) are reported in [7]. Briefly, the results show that each semantic layer contributes positively to performances and using all the layers leads to the best results for all the considered metrics. The URI layer provides the greatest average improvement,[27] although it leads to worse performances for some queries (due to KE errors), while the TIME and FRAME layers consistently provide positive improvements. As shown in Table 5, the TYPE layer is the most widely available layer, followed by URI, TIME and FRAME. Overall, these results show that the injection of semantic information improves the retrieval capabilities of IR systems, and provide interesting cues for understanding the impact of each layer on document ranking and, consequently, on the effectiveness of the approach.

5.1.3. Balancing semantic and textual content

We also evaluated KE4IR using different weights for textual and semantic layers. Figure 3 shows how the NDCG@10 and MAP metrics change when the importance given to the semantic information changes as well. The y-axes report the NDCG@10 (Figure 3a) and MAP (Figure 3b) values, while the x-axis reports the weight $\mathrm{w}(\textsc{semantics})$ assigned to all the semantic information and divided equally among semantic layers, with $\mathrm{w}(\textsc{textual}) = 1 - \mathrm{w}(\textsc{semantics})$; a $\mathrm{w}(\textsc{semantics})$ value of 0.0 means that only textual information is used (and no semantic content), while a value of 1.0 means that only semantic information is used (and no textual content).

The results in Figure 3 show that semantic information impacts positively on system performances up to $w(\textsc{semantics}) \leqslant 0.89$ for NDGC@10 and $w(\textsc{semantics}) \leqslant 0.92$ for MAP, reaching the highest scores ($\text{NDCG@10} = 0.809$, $\text{MAP} = 0.763$) around 0.61 and 0.65, respectively. Similar behaviors can be observed for NDCG and MAP@10. We remark that these scores are better than the ones obtained with equal textual and semantic weights.

We further investigated whether tuning ad-hoc each single layer weight would lead to substantial improvement of the overall performances. We applied both brute-force and learning-to-rank approaches (as implemented in RankLib[28]) to find the layer weight assignments that maximize either NDGC@10 or MAP. However, no substantial improvement (i.e., greater or equal than 0.005) over the optimal scores in Figure 3 was observed. Therefore, for the other evaluations reported next in the paper (Section 5.2 and 5.3), we adopted the weights

$$\mathrm{w}(\textsc{textual}) = 0.35$$
$$\mathrm{w}(\textsc{uri}) = \mathrm{w}(\textsc{type}) = \mathrm{w}(\textsc{time}) = \mathrm{w}(\textsc{frame}) = 0.1625$$

optimizing MAP in Figure 3b.

5.1.4. Comparison with retrieval models in [8, 21]

We compare here the performances of KE4IR against other approaches in the literature that have been evaluated on WES2015 and employ semantic information.[29]

In [8], the authors propose and evaluate some methods exploiting LOD knowledge bases and some variations of the Generalized Vector Space Model. These methods are evaluated only over *annotations manually validated by experts* (i.e., perfect linking to DBpedia entities), so in order to fairly compare the scores reported in [8, Table 4] with KE4IR performances, we removed the URI layer automatically produced by PIKES analysis, replacing it with the manually validated URIs used by [8]. Thus, this KE4IR setting (labelled KE4IR$_{GU}$ to remark that gold URIs were used) leverages manually validated content for the URI layer, the LOD background content (if any) corresponding to these URIs for TYPE and TIME, plus the content automatically obtained via the remaining PIKES analyses (i.e., without entity linking) for the TYPE (as result of WSD), TIME (as result of TERN) and FRAME (as result of SRL) layers.

First, we compare the performance of KE4IR$_{GU}$ against the textual baseline and KE4IR, to better appreciate the benefit of having "correct" entity linking information for the URI layer. The results, reported in Table 6, show that KE4IR$_{GU}$ clearly outperforms both other systems, with an improvement $\geqslant$6% on NDCG@10 and $\geqslant$9% on MAP over the textual base-

[27] Performances of entity linking on queries: precision 96.6%, recall 93.4%, F_1 94.9% (28 correct, 1 incorrect, 2 missing URIs).

[28] http://sourceforge.net/p/lemur/wiki/RankLib/

[29] Beyond the semantic-based approaches considered here, the authors of [38] propose a couple of approaches, one based on Latent Semantic Analysis and one based on automatic query expansion, that do not exploit (symbolic) semantic information at all, and evaluate them on the WES2015 dataset. They apply a different query procedure exploiting some additional assumptions (similar to the one adopted in the Trec 6,7,and 8 Ad-hoc tracks - see Section 5.3.3), and thus their results and ours are not fairly comparable.

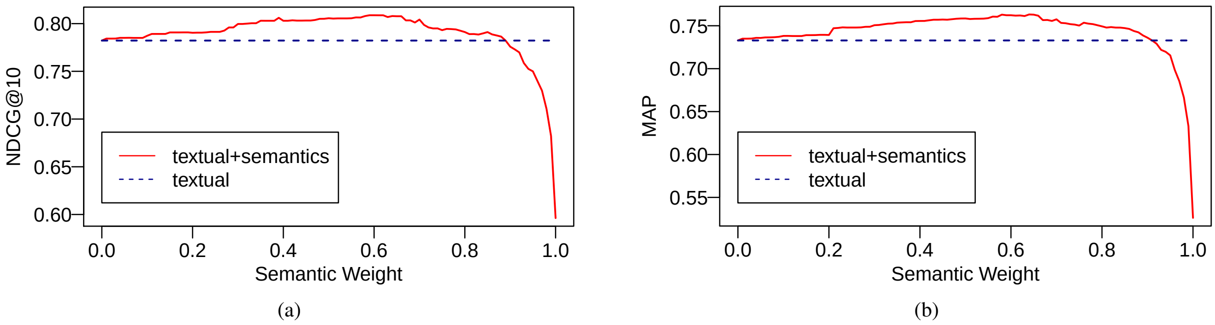


Fig. 3. Trends of (a) NDCG@10 and (b) MAP based on the amount of semantic information considered with respect to the textual one.

line, and an improvement ⩾3% on NDCG@10 and ⩾5% on MAP over KE4IR.

Then, we compare the performance of KE4IR$_{GU}$ against all the semantic methods discussed in [8]. Three main methods are evaluated in [8, Table 4]: (i) *Concept+Text*, that considers URIs of entities as textual terms in building the index in Lucene; (ii) *Connectedness*, that exploits the level of connectedness of entities within documents for computing the relevance score, evaluated in two variants, with and without using term frequency; and, (iii) *Taxonomic*, that leverages taxonomic relationships of the entity classes in the construction of the term vectors, evaluated in two variants, with and without using Resnik similarity.

Table 7 reports the scores in [8, Table 4], together with the relative improvement of KE4IR$_{GU}$ over each method on each measure.[30] KE4IR$_{GU}$ performs consistently better than all the methods in [8]. In particular, KE4IR$_{GU}$ scores similarly to those methods on Prec@1 and NDCG, while it consistently outperforms them on the first 10 documents returned (from 8.49% to 10.51% improvement on NDCG@10, from 23.80% to 32.10% improvement on MAP@10), as well as on MAP (improvement from 4.17% to 12.52%). To complete the comparison, it is interesting to observe (considering the scores in Tables 6 and 7) that also KE4IR (i.e., the standard version relying only on automatically annotated content) outperforms all the methods in [8] on the first 10 documents returned, both in terms of quantity (17.85% to 25.75% improvement on MAP@10) and quality (6.17% to 7.58% improvement on NDCG@10).

In [21], the authors reproduce the results of KE4IR and investigate a family of retrieval models featuring BM25 (and one of its variants) as the similarity measure and URI and TYPE as semantic layers. The latter are used for *conceptual ranking*, by extending the BM25 function, and/or for *semantic filtering*, by excluding the documents not mentioning any of the URI and TYPE terms of the query. Compared to KE4IR on the WES2015 dataset, semantic filtering is shown to substantially improve precision (P@5, P@10) at the expense of other measures, whereas some configurations of conceptual ranking manage to slightly improve MAP and NDCG at the expense of precision. BM25 alone does not provide significant improvements over VSM, whereas the inclusion of URI and TYPE terms from the Wikipedia abstracts of entities appearing in queries and documents does not affect performances. Overall, the results in [21] provide additional evidence that the enrichment of documents and queries with semantic terms may allow improving performances over the use of textual information alone, even when a different retrieval model is used.

5.2. *Fernández et al. 2011 Benchmark for Semantic Search Systems*

In our second evaluation, we assessed the performances of KE4IR on the evaluation dataset proposed in [39] (WT10g collection with F&al. query set in Tables 2 and 3, derived from TREC 9 title and TREC 2001 title) for benchmarking semantic search systems, i.e., systems exploiting semantics- or ontology-based retrieval models.

5.2.1. *Evaluation Set-Up*

As for the WES2015 evaluation, we compared KE4IR against the textual baseline introduced in Section 5 to assess whether the KE-based enrichment of query and document term vectors with semantic terms yields a substantial improvement of IR perfor-

[30] No Prec@5 and Prec@10 scores are reported in [8, Table 4], so we restrict the comparison on Prec@1, NDCG, NDCG@10, MAP, and MAP@10.

Table 6

KE4IR$_{GU}$ performances and comparison with the textual baseline and KE4IR (WES2015 query set and document collection).

Approach/System	**Prec@1**	**Prec@5**	**Prec@10**	**NDCG**	**NDCG@10**	**MAP**	**MAP@10**
Textual	0.943	0.669	0.453	0.832	0.782	0.733	0.681
KE4IR	**0.971**	0.680	0.474	0.854	0.806	0.758	0.713
KE4IR$_{GU}$	**0.971**	**0.691**	**0.482**	**0.879**	**0.831**	**0.800**	**0.749**
KE4IR$_{GU}$ vs. Textual	3.03%	3.42%	6.49%	5.63%	6.18%	9.18%	9.98%
p-value (approx. random.)	0.500	0.135	**0.012**	**0.003**	**0.006**	**0.001**	**0.004**
KE4IR$_{GU}$ vs. KE4IR	0.00%	1.68%	1.86%	2.91%	3.09%	5.49%	5.00%
p-value (approx. random.)	0.500	0.293	0.211	0.065	0.079	**0.011**	**0.039**

Table 7

Comparison of KE4IR$_{GU}$ against all the methods reported in [8] (WES2015 query set and document collection).

Approach/System	**Prec@1**	**NDCG**	**NDCG@10**	**MAP**	**MAP@10**
Concept+Text ($\alpha = 1$)	0.971	0.872	0.761	0.736	0.573
KE4IR$_{GU}$ vs. Concept+Text ($\alpha = 1$)	0.00%	0.80%	9.20%	8.70%	30.72%
Connectedness (only)	0.971	0.862	0.752	0.711	0.567
KE4IR$_{GU}$ vs. Connectedness (only)	0.00%	1.97%	10.51%	12.52%	32.10%
Connectedness (with tf)	0.943	0.874	0.766	0.749	0.583
KE4IR$_{GU}$ vs. Connectedness (with tf)	2.97%	0.57%	8.49%	6.81%	28.47%
Taxonomic (no similarity, $\alpha = 1/2$)	0.943	0.875	0.758	0.766	0.603
KE4IR$_{GU}$ vs. Taxonomic (no similarity, $\alpha = 1/2$)	2.97%	0.46%	9.63%	4.44%	24.21%
Taxonomic (Resnik-Zhou, $\alpha = 1/2$)	0.943	0.877	0.762	0.768	0.605
KE4IR$_{GU}$ vs. Taxonomic (Resnik-Zhou, $\alpha = 1/2$)	2.97%	0.23%	9.06%	4.17%	23.80%

Table 8

Comparison of KE4IR against the textual baseline in dataset of [39] ("F&al." query set, "WT10g" collection).

Approach/System	**Prec@1**	**Prec@5**	**Prec@10**	**NDCG**	**NDCG@10**	**MAP**	**MAP@10**
Textual	0.400	0.420	0.410	0.367	0.344	0.159	0.051
KE4IR	**0.550**	**0.460**	**0.440**	**0.496**	**0.374**	**0.227**	**0.077**
KE4IR vs. Textual	37.50%	9.52%	7.32%	35.14%	8.65%	42.89%	50.56%
p-value (approx. random.)	0.125	0.153	0.268	**0.000**	0.257	**0.001**	0.057

mances (as in Section 5.1). To complement this assessment, we also compared KE4IR performances against the semantic search engine proposed in [9], hereafter named "F&al.", with the goal of understanding whether the simple VSM enriched with semantic terms achieves performances comparable/better/worse than other state-of-the-art semantic-based IR approaches.

5.2.2. Results

Table 8 reports the comparison of KE4IR against the textual baseline. As shown by the results, KE4IR outperforms the textual baseline for all the considered measures. In particular, KE4IR substantially outperforms the textual baseline on Prec@1 (~37%), MAP@10 (~51%), MAP (~43%), and NDCG (~35%), a result, for the latter two measures, statistically significant according to the paired approximate randomization test. Hence, also on this dataset we observe that considering semantic terms enables a substantial improvement of the performances of document retrieval. Again, these results contribute to positively answer our research question.

Table 9 reports the semantic layers exploited by KE4IR (extracted with PIKES) on each query of the dataset. Note that for one query (q457) no semantic information was extracted, mainly due to the plural suffix on Chevrolet; nevertheless, we recall that in these cases, KE4IR resorts to the textual information. For all other queries, TYPE layer content was extracted, while URI content was available for about one third of the queries (7 queries). TIME and FRAME were available only on few queries (2 and 3 queries, respectively).

5.2.3. Comparison with F&al. [9]

Table 9 shows the Prec@10 and MAP results[31] for the comparison of KE4IR against the F&al. semantic search engine proposed in [9], query-by-query. Values

[31] The analysis is restricted to Prec@10 and MAP as only these measures are available for the system presented in [9].

Table 9

Queries, available semantic layers per query, and KE4IR vs F&al. performances (Prec@10 and MAP) on the dataset of [39].

Query		**Layers**				**Prec@10**		**MAP**	
TREC ID	**NL Question**	**URI**	**TYPE**	**TIME**	**FRAME**	**F&al.**	**KE4IR**	**F&al.**	**KE4IR**
q451	Provide information on the Bengal cat breeders.	x	x			0.7	**0.8**	0.42	**0.54**
q452	Describe the habitat for beavers.		x			0.2	**0.7**	0.04	**0.11**
q454	What are the symptoms of Parkinson? What is the treatment for Parkinson?	x	x		x	**0.8**	**0.8**	0.26	**0.42**
q457	Find Chevrolets.					0.1	**0.2**	**0.05**	0.04
q465	What deer diseases can infect humans? What human diseases are transferred by deers?		x			0.3	**0.5**	0.13	**0.18**
q467	Show me all information about Dachshund dog breeders.	x	x			**0.4**	**0.4**	0.10	**0.23**
q476	Show me the movies of Jennifer Aniston.	x	x	x		0.5	**0.6**	0.13	**0.53**
q484	Show me the auto production of Skoda.	x	x	x	x	0.2	**0.5**	0.19	**0.39**
q489	What is the effectiveness of Calcium supplements? What are the benefits of Calcium?		x			0.2	**0.3**	0.09	**0.24**
q491	Show me all tsunamis. Describe disasters produced by tsunamis.		x			**0.2**	0.1	**0.08**	0.06
q494	Show me all members of the rock group Nirvana. What are the members of Nirvana?	x	x		x	**0.9**	0.5	**0.41**	0.16
q504	What is the diet of the manatee?		x			0.2	**0.6**	0.13	**0.40**
q508	Of what diseases hair loss is a symptom? Find diseases for which hair loss is a symptom. What diseases have symptoms of hair loss?		x			**0.5**	0.2	**0.15**	0.07
q511	What diseases does smoking cause? What diseases are caused by smoking?		x			0.4	**0.7**	0.07	**0.30**
q512	How are tornadoes formed? Describe the formation of tornadoes.		x			**0.4**	0.2	**0.25**	0.16
q513	What causes earthquakes? Where do earthquakes occur?		x			0.1	**0.2**	0.08	**0.14**
q516	What is the origin of Halloween? What are the original customs of Halloween?		x			0.1	**0.4**	0.07	**0.16**
q523	How are the clouds formed? Describe the formation of clouds. Explain the process of cloud formation.		x			**0.9**	0.7	**0.29**	0.17
q524	How to erase a scar? How to remove a scar?		x			0.2	**0.3**	0.11	**0.19**
q526	What is BMI?	x	x			**0.1**	**0.1**	**0.09**	0.04

Mean	0.37	**0.44**	0.16	**0.23**
KE4IR vs. F&al.	18.92%		43.75%	
p-value (approx. random.)	0.101		**0.026**	

in bold correspond to the best results for the corresponding metric and query (also known as "question" in the considered dataset). KE4IR outperforms F&al. both on Prec@10 and MAP, scoring respectively 0.44 and 0.23 on average on all queries. This accounts for a ~19% and a ~44% increase in performance, the latter statistically significant according to the paired approximate randomization test. For Prec@10, KE4IR provides better results than the other semantic-based approach on 11 queries (55%) and equal results for other 3 queries (15%). For MAP, KE4IR provides better results than F&al. on 13 queries (65%).

Interestingly, in the single case where all semantic layers were available, KE4IR substantially outperforms the other approach, as well as in all the cases where the TIME layer was available. Considering only queries containing some URIs, KE4IR outperforms the other approach in 5 of the 7 cases, performing worse in q494 (likely due to "Nirvana" wrongly linked to DBpedia entity dbpedia:Nirvana rather than dbpedia:Nirvana_(band)) and q526 (likely due to "BMI" wrongly linked to DBpedia entity dbpedia:Broadcast_Music,_Inc. rather than dbpedia:Body_mass_index).[32]

For the sake of completeness, we recall that in [9] the F&al. approach is also compared against two keyword-based approaches, namely Lucene and the best TREC automatic system (i.e., the system achieving the best performances on MAP), using as input queries the original titles of the TREC 9 and TREC 2001 Web Track topics from which the 20 F&al. queries were derived from. Lucene achieved 0.25 and 0.10 on Prec@10 and MAP respectively, while the best TREC automatic system scored 0.3 and 0.2 for the same measures. All these values are substantially lower that the performances of KE4IR reported in Table 9.

5.3. Keyword-based Datasets: TREC 6-7-8-9-2001

In this third evaluation of KE4IR performances, we applied our approach on some standard IR datasets

[32] Performances of entity linking on queries: precision 71.4%, recall 62.5%, F_1 66.7% (5 correct, 2 incorrect, 3 missing URIs).

from TREC typically used for evaluating keyword-based approaches (see TREC document collections and query sets in Tables 2 and 3). In particular, these datasets were adopted as benchmark in the Ad-hoc track of TREC 6, 7, and 8, and the Web track of TREC 9 and 2001.

5.3.1. Evaluation Set-Up

Similarly to the previous two evaluations, we compared KE4IR against the textual baseline. First, we compared the two approaches considering as input queries the *titles* of the TREC topics, which resemble the queries users typically fire to a search engine. This is the typical and widely-accepted evaluation setting, also recommended by TREC guideline documents (e.g., [35]). Then, to collect more evidences for our assessment, we also investigated the query variant comprising the topic title together with the corresponding *description* provided by TREC, a short text explaining the expected results of the query. The following query (q311) from TREC 6 highlights the difference between title and description:

<id>q311**</id>**

<title>Industrial Espionage**</title>**

<desc>Document will discuss the theft of trade
secrets along with the sources of information:
trade journals, business meetings, data from
Patent Offices, trade shows, or analysis of a
competitor's products.**</desc>**

Topic title usually consists of one or more keywords (on average 3 ± 1.5 tokens), sometimes forming a multi-word expression (like q311 title) or short phrase, while description usually consists of well-formed and verbose sentences (on average 14 ± 8 tokens, longest one reaching 62), explaining the criteria for identifying relevant documents, and frequently involving several words that do not characterize the query (e.g., "Documents will discuss...", "Find/Identify documents that...", "Give examples of", "Provide information..."). That is, the description is not a query per se, but rather an explanation of how to answer the query.

Finally, similarly to what done for the F&al. benchmark, we complemented this assessment by analyzing KE4IR performances in light of the best performing systems (with respect to MAP and Prec@10) that participated in the TREC 6, 7, 8, 9, and 2001 competitions where the considered TREC datasets were used, with the goal of understanding whether the simple VSM enriched with semantic terms achieves performances comparable/better/worse than other state-of-the-art IR approaches, typically exploiting more elaborated and performing retrieval models than VSM.

5.3.2. Results

Table 10 reports the comparison of KE4IR against the textual baseline, for the various TREC datasets and considering the query variant consisting of the topic title only. Statistically significant results are highlighted in bold.

As shown by the results, KE4IR outperforms the textual baseline in 33 out of the 35 measures (5 datasets, 7 measures each) reported in Table 10, the only exceptions being Prec@1 on TREC 8 and MAP@10 on TREC 9. In particular, KE4IR performs substantially better than the textual baseline on Prec@10 (relative increments from 0.52% to 9.52%), NDCG (from 1.51% to 22.20%), NDCG@10 (from 0.34% to 12.18%), and MAP (from 0.89% to 38.97%). NDCG and MAP results, which assess the whole rankings produced by the system, are statistically significant in half of the cases. The improvement on NDCG@10, which consider the first 10 ranking positions, is significant for TREC 2001, possibly benefiting from the use of fine-grained 3-value relevance judgments. Overall, also on all these datasets (summing up to 250 queries and 2.2M documents) we observe that considering semantic terms substantially improves IR performances.

We continue our analysis comparing KE4IR against the textual baseline on the various TREC datasets, considering the query variant that concatenates the topic titles with the corresponding descriptions. Table 11 reports the results (statistical significant ones are highlighted in bold). As shown by the results, KE4IR outperforms the textual baseline in 34 out of the 35 measures reported in Table 11, the only exception being Prec@1 on TREC 2001. In particular, KE4IR performs substantially better than the textual baseline on Prec@10 (from 0.69% to 18.18%), NDCG (from 18.64% to 48.99%), NDCG@10 (from 3.80% to 21.25%), MAP (from 18.67% to 63,47%), and MAP@10 (from 8.49% to 34.44%). Improvements on MAP and NDCG are always statistically significant in the considered datasets, whereas the improvements on the other measures are significant in 10 cases out of 25 (5 datasets, 5 measures each). Overall, also for this query variant, we observe that considering semantic terms improves retrieval performances.

Table 10

KE4IR vs textual baseline on all the TREC datasets considered (query = Topic Title only).

Dataset	Approach/System	Prec@1	Prec@5	Prec@10	NDCG	NDCG@10	MAP	MAP@10
TREC 6 title	Textual	0.380	0.388	0.308	0.419	0.357	0.189	0.091
	KE4IR	0.420	0.392	0.322	0.446	0.369	0.203	0.095
	KE4IR vs. Textual	10.53%	1.03%	4.49%	6.49%	3.32%	7.81%	4.40%
	p-value (approx. random.)	0.314	0.438	0.301	**0.027**	0.316	0.156	0.255
TREC 7 title	Textual	0.480	0.400	0.378	0.408	0.397	0.169	0.065
	KE4IR	0.500	0.420	0.394	0.432	0.411	0.182	0.069
	KE4IR vs. Textual	4.17%	5.00%	4.23%	6.02%	3.56%	7.41%	5.05%
	p-value (approx. random.)	0.500	0.129	0.164	**0.001**	0.195	**0.021**	0.062
TREC 8 title	Textual	0.540	0.416	0.386	0.450	0.413	0.190	0.074
	KE4IR	0.500	0.428	0.388	0.457	0.414	0.192	0.075
	KE4IR vs. Textual	-7.41%	2.88%	0.52%	1.51%	0.34%	0.89%	1.84%
	p-value (approx. random.)	0.250	0.266	0.407	0.196	0.465	0.346	0.336
TREC 9 title	Textual	0.360	0.284	0.267	0.393	0.302	0.176	0.110
	KE4IR	0.360	0.292	0.284	0.412	0.309	0.184	0.102
	KE4IR vs. Textual	0.00%	2.82%	6.11%	4.89%	2.49%	4.75%	-7.01%
	p-value (approx. random.)	0.500	0.388	0.096	0.127	0.317	0.322	0.376
TREC 2001 title	Textual	0.380	0.356	0.294	0.336	0.289	0.122	0.053
	KE4IR	0.400	0.388	0.322	0.410	0.325	0.170	0.067
	KE4IR vs. Textual	5.26%	8.99%	9.52%	22.20%	12.18%	38.97%	25.95%
	p-value (approx. random.)	0.500	0.060	0.051	**0.000**	**0.020**	**0.000**	0.059

Table 11

KE4IR vs textual baseline on all the TREC datasets considered (query = Topic Title + Description).

Dataset	Approach/System	Prec@1	Prec@5	Prec@10	NDCG	NDCG@10	MAP	MAP@10
TREC 6 desc	Textual	0.400	0.312	0.264	0.286	0.294	0.118	0.067
	KE4IR	0.440	0.368	0.312	0.389	0.356	0.176	0.089
	KE4IR vs. Textual	10.00%	17.95%	18.18%	35.97%	21.25%	48.99%	34.44%
	p-value (approx. random.)	0.250	**0.016**	**0.009**	**0.000**	**0.004**	**0.000**	0.050
TREC 7 desc	Textual	0.460	0.388	0.342	0.312	0.366	0.114	0.052
	KE4IR	0.480	0.416	0.402	0.394	0.421	0.162	0.065
	KE4IR vs. Textual	4.35%	7.22%	17.54%	26.38%	15.05%	41.64%	23.44%
	p-value (approx. random.)	0.500	0.158	**0.006**	**0.000**	**0.004**	**0.000**	**0.000**
TREC 8 desc	Textual	0.480	0.400	0.368	0.354	0.395	0.150	0.071
	KE4IR	0.500	0.408	0.382	0.420	0.410	0.178	0.077
	KE4IR vs. Textual	4.17%	2.00%	3.80%	18.64%	3.80%	18.67%	8.49%
	p-value (approx. random.)	0.500	0.409	0.127	**0.000**	0.055	**0.000**	0.102
TREC 9 desc	Textual	0.380	0.328	0.288	0.270	0.275	0.118	0.081
	KE4IR	0.440	0.340	0.290	0.403	0.323	0.184	0.104
	KE4IR vs. Textual	15.79%	3.66%	0.69%	48.99%	17.61%	56.63%	28.26%
	p-value (approx. random.)	0.124	0.304	0.473	**0.000**	**0.025**	**0.000**	**0.010**
TREC 2001 desc	Textual	0.440	0.380	0.296	0.266	0.294	0.099	0.058
	KE4IR	0.400	0.388	0.332	0.389	0.343	0.163	0.072
	KE4IR vs. Textual	-9.09%	2.11%	12.16%	46.28%	16.52%	63.47%	23.29%
	p-value (approx. random.)	0.250	0.321	**0.032**	**0.000**	**0.004**	**0.000**	0.058

The comparison of Tables 10 and 11 shows that in general the textual baseline performs worse in the query variant considering both topic title and description. This suggests that the extra *textual* information in the description introduces mainly noise, which is unsurprising given the nature and typical content of topic descriptions (cf. q311 example and following discussion at the beginning of Section 5.3.1). Evaluated on the same topic title and description query variant, KE4IR performs worse on MAP and NDCG measures computed on the whole ranking, but comparably or even better on Prec@10, NDCG@10 and MAP@10 measures computed on the top documents returned. This suggests that the *semantic* information extracted from descriptions helps in identifying highly relevant documents, although it also introduces noise in the long tail of the produced document rankings.

To conclude, the results with both query variants contribute to positively answer our evaluation research question.

5.3.3. *Comparison with best performing systems at TREC 6-7-8-9-2001*

To better put KE4IR performances in perspective, we complement our analysis by reporting the results of the best performing systems that participated in the considered TREC evaluation campaigns.

Before proceeding, note that KE4IR can be fairly compared only with the results of the systems that participated at the Web track of the TREC 9 and 2001 editions. In fact, in the TREC 6, 7, and 8 Ad-hoc tracks, a specific query procedure was adopted: a preliminary query expansion process was applied, leveraging the assumption that the document collection is *static*, and *known beforehand* to the system designers. However, as this setting does not mimic what a user actually does when using a real-world search engine, the Ad-hoc track evolved in TREC 9 and 2001 into the Web track, where queries are directly compared with the document collection, to more realistically simulate the interaction between users and search engines, especially in a web-like setting. This latter query procedure is the same we followed in our approach.

Table 12 reports the Prec@10 and the MAP values obtained by the best Prec@10 and by the best MAP performers, together with KE4IR results, when using topic titles as queries. TREC 6, 7, and 8 system performances are in italics, to remark the different query procedure adopted in the Ad-Hoc track. As expected, the ad-hoc TREC systems, specifically designed and highly tuned for working with static document collections and with the possibility of performing a preliminary query execution run, obtained better results with respect to performances of the web-like query procedure implemented in KE4IR. However, when the possibility of performing the preliminary screening is not allowed (TREC 9 and 2001), the effectiveness of KE4IR is closer to the ones of the best systems, achieving the highest Prec@10 score on TREC 9 title. For completeness, Table 13 reports the Prec@10 and the MAP values obtained by the best Prec@10 and by the best MAP performers, together with KE4IR results, when using also descriptions, concatenated to the corresponding topic titles, as queries.

This suggests that enriching the simple textual-term based VSM with semantic terms, enables to achieve performances comparable to systems employing more sophisticated retrieval models, thus further corroborating our hypothesis that semantic content can positively affect document retrieval performances.

6. Discussion

In this section, to further elaborate on the investigation of the research question introduced at the beginning of Section 5, we globally analyze the performances of KE4IR versus the textual baseline across the three evaluations conducted.[33] First of all, we observe that KE4IR performs constantly better than the textual baseline on all the datasets and for all the measures considered (with the only exception of Prec@1 on TREC 8, and MAP@10 on TREC 9). The improvements on MAP (up to 42.89%) are statistically confirmed in 4 cases out of 7, while for NDCG (up to 35.14%) statistically significant improvements are obtained in 5 cases out of 7. In particular, for NDCG (a measure particularly useful in multi-value relevance scenarios) and its first ten documents variant (NDCG@10), the improvement is statistically significant on both experiments involving multiple relevance levels (WES2015 and TREC 2001), thus suggesting that semantic content may effectively improve IR performances in multi-value settings.

Although in almost all cases we observe an improvement, there is high variability among the evaluation datasets in terms of relative improvements with respect to the textual baseline. The higher improvement on F&al. may be due to the possibility to extract and exploit more structure in F&al. queries, which are longer than WES2015 ones. Concerning the WES2015 dataset, it must be noted that the performances of the textual baseline are significantly higher here (0.733 MAP, 0.782 NDCG@10) than in other datasets (0.099-0.190 MAP, 0.289-0.413 NDCG@10), so the relative improvement of KE4IR is inevitably smaller for WES2015 compared to other datasets. Conversely, the absolute improvements exhibit less variability across the datasets, with the absolute improvements on WES2015 (0.025 MAP, 0.024 NDCG@10) well within the range of improvements observed for other datasets (0.002-0.068 MAP, 0.001-0.062 NDCG@10).

To better understand how KE4IR compares with the textual baseline across the different evaluation datasets, we consider each ⟨query, document collection⟩ pair as one subject, assessing performances of both

[33] For the sake of comparing "similar" queries, for Section 5.3 we consider only the results for the TREC title query variants.

Table 12

KE4IR vs. best Prec@10 and MAP systems (considered separately) in corresponding TREC campaign (query = Topic Title only).

Dataset	**Precision at 10 documents**			**Mean Average Precision**		
	Best System Prec@10	Best System MAP	KE4IR	Best System Prec@10	Best System MAP	KE4IR
TREC 6	*0.438*	*0.438*	0.322	*0.288*	*0.288*	0.203
TREC 7	*0.486*	*0.486*	0.394	*0.261*	*0.261*	0.182
TREC 8	*0.488*	*0.408*	0.388	*0.279*	*0.306*	0.192
TREC 9	0.276	0.238	**0.284**	0.179	**0.201**	0.184
TREC 2001	**0.344**	**0.344**	0.322	**0.223**	**0.223**	0.170

Table 13

KE4IR vs. best Prec@10 and best MAP systems (considered separately) in corresponding TREC campaign (query = Topic Title + Description). Note that for TREC 6 (*) we include the results of the best runs of type “Category A, Automatic, Long”, which beyond topic title and description might have used also the additional (noisy) field “narrative”.

Dataset	**Precision at 10 documents**			**Mean Average Precision**		
	Best System Prec@10	Best System MAP	KE4IR	Best System Prec@10	Best System MAP	KE4IR
TREC 6*	*0.464*	*0.448*	0.312	*0.231*	*0.260*	0.176
TREC 7	*0.526*	*0.526*	0.402	*0.281*	*0.281*	0.162
TREC 8	*0.550*	*0.508*	0.382	*0.317*	*0.321*	0.178
TREC 9	**0.350**	0.238	0.338	0.229	**0.262**	0.184
TREC 2001	**0.446**	**0.446**	0.332	**0.233**	**0.233**	0.163

systems over the collection of all subjects from the three evaluations. This boils down to compare KE4IR and the textual baseline performances over a set of 305 subjects. Here our purpose is not to derive overall, absolute measures per se, as different query sets and document collections were involved, but rather to understand if on all the queries served by KE4IR and the textual baseline, for all the evaluation measures considered, there is a substantial difference in the distribution of the scores of the two systems. Table 14 summarizes the results of this analysis. The statistical significance analysis shows that, for each measure with the exception of Prec@1, the distribution of KE4IR scores on ⟨query, document collection⟩ pairs is higher than the distribution of the textual baseline scores, thus further confirming that KE4IR performs consistently better than the textual baseline. A complementary view on the same data is provided by Figure 4, which shows, for each measure, the boxplots of the score differences between KE4IR and the textual baseline. Besides mean values (cross marks) and medians (bold line), first (lower hinge) and third quartiles (upper hinge), and 95% confidence interval of medians (whiskers) are shown. Outliers (outside the 95% confidence) are not shown. Mean difference values are all above zero (as confirmed also by Table 14), while median values tend to be close to zero (with the notable exception of NDCG and MAP). This suggests that while in many queries the difference is minimal or absent, there are a number of queries for which KE4IR performances substantially improve over the textual baseline. Note that the lower 25% hinge is around zero in all cases, showing that most of the differences (~75%) are non-negative. Furthermore, for NDGG and MAP, more than 50% of the differences have values strictly greater than zero. Moreover, the top whisker is longer than the bottom one in most cases, thus suggesting that for queries for which there is positive difference between KE4IR and the textual baseline, the absolute value of the difference is generally larger that the cases where a negative value occurs.

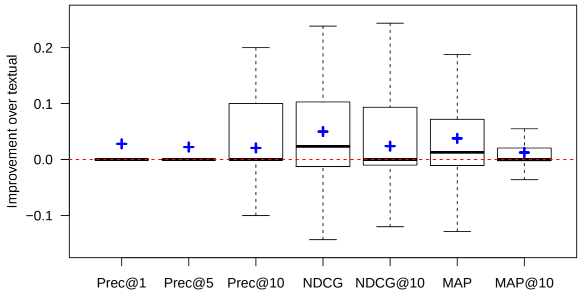


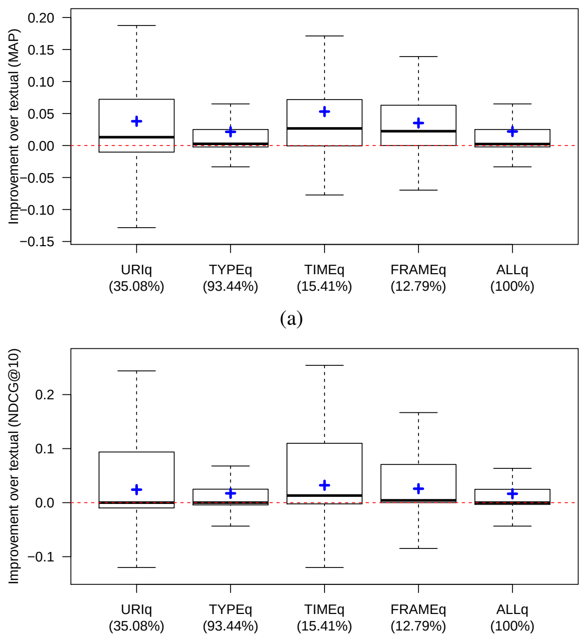
Fig. 4. Boxplot of KE4IR increments over textual baselines on each query, for all measures.

To get more insights on the data collected, we also comparatively analyzed KE4IR and textual baseline performances according to the semantic content identified in the queries. That is, for each semantic layer (layers URI, TYPE, TIME, FRAME), we considered the queries containing at least one semantic term from that layer, and we analyzed the distribution of the score differences between KE4IR and the textual baseline on

Table 14

KE4IR and the textual baseline over all queries.

Approach/System	**Prec@1**	**Prec@5**	**Prec@10**	**NDCG**	**NDCG@10**	**MAP**	**MAP@10**
Textual	0.485	0.407	0.347	0.448	0.401	0.233	0.146
KE4IR	**0.505**	**0.422**	**0.363**	**0.484**	**0.417**	**0.255**	**0.154**
KE4IR vs. Textual	4.05%	3.87%	4.81%	8.08%	4.10%	9.36%	5.68%
p-value (approx. random.)	0.153	**0.019**	**0.008**	**0.000**	**0.006**	**0.000**	**0.010**



(a)

(b)

Fig. 5. (a) MAP and (b) NDCG@10 boxplots for different subsets of queries based on the semantic content they contain. Cross marks represent mean values.

these queries (respectively, query sets URI$_q$, TYPE$_q$, TIME$_q$, FRAME$_q$). Figures 5a and 5b show the boxplots for MAP and NDCG@10, respectively; for reference, also the boxplot of the same measure on all queries is also shown (ALL$_q$). Percentages below each set indicate the relative size of the query set considered with respect to the total number of queries. Given the size, the distributions for the TYPE$_q$ query set clearly resemble the one considering all queries. The other three query sets, covering more selective subsets of queries, show greater variability, as highlighted by the longer whiskers and quartile boxes. Mean values are always higher than medians, and most part of each distribution, 75% for TIME$_q$ and FRAME$_q$ while 50% for URI$_q$, are in the positive half of the plots, confirming that for many queries, when URI, TIME and FRAME content is available, the semantic analysis enables to achieve higher scores than just using raw textual terms. The longer top and bottom whiskers for URI$_q$ remark the impact of including URI layer content, and the importance of the entity linking analysis: if entities are correctly linked, this may lead to a substantial boost of performances, while linking a span of text to the wrong entity (e.g., wrongly linking "BMI" to DBpedia entity dbpedia:Broadcast_Music,_Inc. instead of dbpedia:Body_mass_index) may kill performances. Similar considerations hold for TIME$_q$ (e.g., adding time information from wrongly linked entities) and FRAME$_q$ (e.g., adding frame-role pairs involving wrongly linked entities).

In [7], we investigated the performances of using different layer combinations in KE4IR, using only WES2015 query set and document collection. We rerun the same analysis but considering all the ⟨query, document collection⟩ pairs from the various evaluations. The boxplots for MAP are shown in Figure 6. Combining all the semantic layers produces the best performances for all the considered metrics, as confirmed by the mean difference value on MAP of 0.0218 over the textual baselines (second best combination stops at 0.0216). All the eight combinations containing TYPE produce better results than their corresponding configurations that do not consider that layer. Among those exploiting TYPE, each of the four combinations exploiting also URI produces better results than its corresponding configuration that does not consider that layer. Adding TIME and FRAME further improves the scores. These results, in line with the findings in [7], confirm that the integration of different semantic information leads to a general improvement of the effectiveness of the document retrieval task. Similar consideration can be drawn also for the other measures.

7. Concluding Remarks and Future Work

In this paper, we investigated the benefits of exploiting knowledge extraction techniques and background knowledge from LOD sources to improve the effectiveness of document retrieval systems. We developed

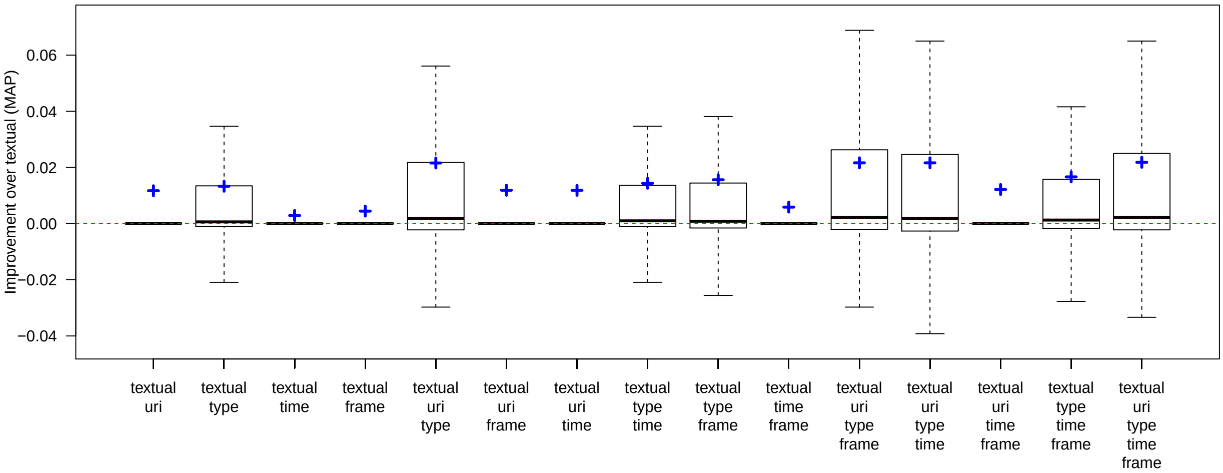


Fig. 6. MAP boxplots of KE4IR increments over the textual baselines, using different layer combinations.

an approach, called KE4IR, where queries and documents are enriched with semantic content such as entities, types, semantic frames, and temporal information, automatically obtained by processing their text with a state-of-the-art knowledge extraction tool (PIKES [3]). Relevance of documents for a given query is computed using an adaptation of the well-known Vector Space Model, on query and document vectors comprising both semantic content and textual terms.

We evaluated KE4IR on several state-of-the-art document retrieval datasets, on a total of 555 queries and with document collections spanning from few hundreds to more than a million of resources. Besides showing the feasibility of applying knowledge extraction techniques for document retrieval on large, web-like collections (e.g., WT10g), the results on all datasets show that complementing the textual information of queries and documents with the semantic content extracted from them enables to consistently outperform approaches using only textual information, further validating similar conclusions drawn by previous works (e.g., [7–9]). Furthermore, on a specifically devised dataset for semantic search, KE4IR achieves scores substantially higher than another reference ontology-based search system [9].

The comprehensive assessment performed, substantially extending the preliminary results presented in [7], gave interesting insights about the application of automatic knowledge extraction techniques for document retrieval, highlighting aspects on which to work on for augmenting the effectiveness of the retrieval system. For instance, wrong entity linking analysis on a query may lead to poor overall results. While some of these aspects will be addressed by the progress and new achievements in knowledge extraction, the use of approaches favoring precision of extracted content instead of recall may be a winning strategy for obtaining a better average improvement of system effectiveness.

So far, KE4IR performances were assessed in general knowledge context, and in the implementation we considered only general-purpose knowledge bases for enriching documents. Further investigations we plan to conduct involve applying KE4IR on domain-specific contexts, such as the biomedical one (e.g., BIOASQ challenge[34]). While changes in the architectural grounds of the approach are not foreseen, this could require the use of domain-specific resources able to provide more effective annotations: e.g., domain-specific KBs such as SNOMED CT[35] can be used with KB-agnostic entity linking tools to extract domain-specific URI and TYPE terms; for FRAME terms, domain-specific frames can be defined and annotated in a corpus to retrain the semantic role labeling tools used for knowledge extraction.

Further extensions of the work may consider to include the confidence score returned by the NLP tools (e.g., Stanford CoreNLP NERC, DBpedia Spotlight) for the extracted content when computing the weight of the indexed terms, as well as the adoption of a more sophisticated retrieval model. The latter may consist either in a different variation of VSM, e.g., featuring a soft-cosine similarity measure [40] in place of the sim-

[34] http://www.bioasq.org/participate/challenges
[35] http://b2i.sg/

ple dot product used now, or in the extension of a more advanced state-of-the-art retrieval model, in line with the investigation conducted in [21], where KE4IR content was used with a BM25 model. Moreover, by considering the semantic layers we included in our model, we may consider to expand the information brought by the TYPE layer by including the Information Content associated with each entity included in such a layer [41]. Finally, we may consider the integration of a further layer including information computed by an embeddings-based analysis of the other layers considered for describing document content. This new perspective would open the possibility of detecting different kind of similarities between documents and queries with respect to the implemented ones.

Acknowledgments We would like to thank Alessio Palmero Aprosio for his contribution to the implementation of KE4IR, and the authors of [8] and [9] for their helpfulness in answering our questions. We also thank the reviewers and editor for their many insightful comments and suggestions that helped us improving the manuscript.

References

[1] F. Draicchio, A. Gangemi, V. Presutti and A.G. Nuzzolese, FRED: From Natural Language Text to RDF and OWL in One Click, in: *ESWC 2013 Satellite Events, Revised Selected Papers*, LNCS, Vol. 7955, Springer, 2013, pp. 263–267.

[2] P. Vossen, R. Agerri, I. Aldabe, A. Cybulska, M. van Erp, A. Fokkens, E. Laparra, A.-L. Minard, A.P. Aprosio, G. Rigau, M. Rospocher and R. Segers, NewsReader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news, *Knowl.-Based Syst.* **110** (2016), 60–85.

[3] F. Corcoglioniti, M. Rospocher and A. Palmero Aprosio, Frame-Based Ontology Population with PIKES, *IEEE Trans. Knowl. Data Eng.* **28**(12) (2016), 3261–3275, ISSN 1041-4347.

[4] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer, DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic Web* **6**(2) (2015), 167–195.

[5] J. Hoffart, F.M. Suchanek, K. Berberich and G. Weikum, YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia, *Artif. Intell.* **194** (2013), 28–61, ISSN 0004-3702.

[6] G. Salton, A. Wong and C.S. Yang, A Vector Space Model for Automatic Indexing, *Commun. ACM* **18**(11) (1975), 613–620, ISSN 0001-0782.

[7] F. Corcoglioniti, M. Dragoni, M. Rospocher and A. Palmero Aprosio, Knowledge Extraction for Information Retrieval, in: *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*, H. Sack, E. Blomqvist, M. d'Aquin, C. Ghidini, S.P. Ponzetto and C. Lange, eds, Lecture Notes in Computer Science, Vol. 9678, Springer, 2016, pp. 317–333. http://dx.doi.org/10.1007/978-3-319-34129-3_20.

[8] J. Waitelonis, C. Exeler and H. Sack, Linked Data Enabled Generalized Vector Space Model To Improve Document Retrieval, in: *Proc. of NLP and DBpedia Workshop*, CEUR Workshop Proceedings, Vol. 1581, CEUR-WS.org, pp. 33–44.

[9] M. Fernández, I. Cantador, V. Lopez, D. Vallet, P. Castells and E. Motta, Semantically enhanced Information Retrieval: An ontology-based approach, *J. Web Sem.* **9**(4) (2011), 434–452.

[10] W.B. Croft, User-Specified Domain Knowledge for Document Retrieval, in: *Proc. of SIGIR*, ACM, 1986, pp. 201–206. ISBN 0-89791-187-3.

[11] J. Gonzalo, F. Verdejo, I. Chugur and J. Cigarran, Indexing with WordNet synsets can improve text retrieval, in: *Usage of WordNet in Natural Language Processing Systems*, 1998. http://www.aclweb.org/anthology/W98-0705.

[12] C. Fellbaum (ed.), *WordNet: An Electonic Lexical Database*, MIT Press, 1998.

[13] M. Dragoni, C. da Costa Pereira and A. Tettamanzi, A conceptual representation of documents and queries for Information Retrieval systems by using light ontologies, *Expert Syst. Appl.* **39**(12) (2012), 10376–10388.

[14] O. Dridi, Ontology-based Information Retrieval: Overview and new proposition, in: *Proc. of Int. Conf. on Research Challenges in Information Science (RCIS)*, 2008, pp. 421–426, ISSN 2151-1349.

[15] S.L. Tomassen, Research on Ontology-Driven Information Retrieval, in: *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, R. Meersman, Z. Tari and P. Herrero, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 1460–1468. ISBN 978-3-540-48276-5.

[16] P. Castells, M. Fernandez and D. Vallet, An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval, *IEEE Trans. Knowl. Data Eng.* **19**(2) (2007), 261–272, ISSN 1041-4347.

[17] D. Vallet, M. Fernández and P. Castells, An Ontology-Based Information Retrieval Model, in: *The Semantic Web: Research and Applications*, A. Gómez-Pérez and J. Euzenat, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 455–470. ISBN 978-3-540-31547-6.

[18] R.Y.K. Lau, C.C.L. Lai and Y. Li, Mining Fuzzy Ontology for a Web-Based Granular Information Retrieval System, in: *Proc. of Int. Conf on Rough Sets and Knowledge Technology (RSKT)*, LNCS, Vol. 5589, Springer, 2009, pp. 239–246. ISBN 978-3-642-02962-2.

[19] M. Sy, S. Ranwez, J. Montmain, A. Regnault, M. Crampes and V. Ranwez, User centered and ontology based information retrieval system for life sciences, *BMC Bioinformatics* **13**(S–1) (2012), 4. doi:10.1186/1471-2105-13-S1-S4. https://doi.org/10.1186/1471-2105-13-S1-S4.

[20] A. Jimeno-Yepes, R. Berlanga Llavori and D. Rebholz-Schuhmann, Ontology refinement for improved Information Retrieval, *Inf. Process. Manage.* **46**(4) (2010), 426–435, ISSN 0306-4573.

[21] J. Azzopardi, F. Benedetti, F. Guerra and M. Lupu, Back to the Sketch-Board: Integrating Keyword Search, Semantics, and Information Retrieval, in: *Semantic Keyword-Based*

Search on Structured Data Sources - COST Action IC1302 Second International KEYSTONE Conference, IKC 2016, LNCS, Vol. 10151, 2016, pp. 49–61.

[22] A. Spink, B.J. Jansen, C. Blakely and S. Koshman, A study of results overlap and uniqueness among major web search engines, *Inf. Process. Manage.* **42**(5) (2006), 1379–1391, ISSN 0306-4573.

[23] N. Stojanovic, An Approach for Defining Relevance in the Ontology-Based Information Retrieval, in: *Proc. of IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI)*, IEEE Computer Society, 2005, pp. 359–365. ISBN 0-7695-2415-X.

[24] M. Baziz, M. Boughanem, G. Pasi and H. Prade, An Information Retrieval Driven by Ontology: from Query to Document Expansion, in: *Large Scale Semantic Access to Content (Text, Image, Video, and Sound) (RIAO)*, 2007, pp. 301–313.

[25] F. Corcoglioniti, M. Rospocher and A.P. Aprosio, A 2-phase Frame-based Knowledge Extraction Framework, in: *Proc of ACM Symposium on Applied Computing (SAC)*, ACM, 2016, pp. 354–361. ISBN 978-1-4503-3739-7.

[26] J. Rouces, G. de Melo and K. Hose, FrameBase: Representing N-Ary Relations Using Semantic Frames, in: *The Semantic Web. Latest Advances and New Domains*, F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux and A. Zimmermann, eds, Springer International Publishing, Cham, 2015, pp. 505–521. ISBN 978-3-319-18818-8.

[27] K. Höffner, S. Walter, E. Marx, R. Usbeck, J. Lehmann and A.N. Ngomo, Survey on challenges of Question Answering in the Semantic Web, *Semantic Web* **8**(6) (2017), 895–920. doi:10.3233/SW-160247. https://doi.org/10.3233/SW-160247.

[28] V. López, V.S. Uren, M. Sabou and E. Motta, Is Question Answering fit for the Semantic Web?: A survey, *Semantic Web* **2**(2) (2011), 125–155. doi:10.3233/SW-2011-0041. https://doi.org/10.3233/SW-2011-0041.

[29] S. Hazrina, N.M. Sharef, H. Ibrahim, M.A.A. Murad and S.A.M. Noah, Review on the advancements of disambiguation in semantic question answering system, *Inf. Process. Manage.* **53**(1) (2017), 52–69. doi:10.1016/j.ipm.2016.06.006. https://doi.org/10.1016/j.ipm.2016.06.006.

[30] I.O. Mulang, K. Singh and F. Orlandi, Matching Natural Language Relations to Knowledge Graph Properties for Question Answering, in: *Proceedings of the 13th International Conference on Semantic Systems, SEMANTICS 2017, Amsterdam, The Netherlands, September 11-14, 2017*, R. Hoekstra, C. Faron-Zucker, T. Pellegrini and V. de Boer, eds, ACM, 2017, pp. 89–96. doi:10.1145/3132218.3132229. http://doi.acm.org/10.1145/3132218.3132229.

[31] C. da Costa Pereira, M. Dragoni and G. Pasi, Multidimensional relevance: Prioritized aggregation in a personalized Information Retrieval setting, *Inf. Process. Manage.* **48**(2) (2012), 340–357, ISSN 0306-4573.

[32] C.D. Manning, P. Raghavan, H. Schütze et al., *Introduction to Information Retrieval*, Vol. 1, Cambridge University Press, 2008. ISBN 978-0-521-86571-5.

[33] F. Corcoglioniti, M. Rospocher, M. Mostarda and M. Amadori, Processing Billions of RDF Triples on a Single Machine using Streaming and Sorting, in: *Proc. of ACM Symposium on Applied Computing (SAC)*, ACM, 2015, pp. 368–375. ISBN 978-1-4503-3196-8.

[34] K. Järvelin and J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, *ACM Trans. Inf. Syst.* **20**(4) (2002), 422–446, ISSN 1046-8188.

[35] E.M. Voorhees and D. Harman, Overview of the Sixth Text REtrieval Conference (TREC-6), Pergamon Press, Inc., Tarrytown, NY, USA, 2000, pp. 3–35, ISSN 0306-4573. doi:10.1016/S0306-4573(99)00043-6. http://dx.doi.org/10.1016/S0306-4573(99)00043-6.

[36] E.W. Noreen, *Computer-intensive methods for testing hypotheses: an introduction*, Wiley, 1989. ISBN 9780471611363.

[37] M.D. Smucker, J. Allan and B. Carterette, A Comparison of Statistical Significance Tests for Information Retrieval Evaluation, in: *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, ACM, New York, NY, USA, 2007, pp. 623–632. ISBN 978-1-59593-803-9. doi:10.1145/1321440.1321528. http://doi.acm.org/10.1145/1321440.1321528.

[38] C. Layfield, J. Azzopardi and C. Staff, Experiments with Document Retrieval from Small Text Collections Using Latent Semantic Analysis or Term Similarity with Query Coordination and Automatic Relevance Feedback, in: *Semantic Keyword-Based Search on Structured Data Sources - COST Action IC1302 Second International KEYSTONE Conference, IKC 2016*, LNCS, Vol. 10151, 2016, pp. 25–36.

[39] M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta and P. Castells, Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale, in: *Semantic Search 2009 Workshop at the 18th International World Wide Web Conference (WWW 2009)*, 2009. http://oro.open.ac.uk/23481/.

[40] G. Sidorov, A.F. Gelbukh, H. Gómez-Adorno and D. Pinto, Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model, *Computación y Sistemas* **18**(3) (2014).

[41] S. Harispe, S. Ranwez, S. Janaqi and J. Montmain, Semantic Similarity from Natural Language and Ontology Analysis, *CoRR* **abs/1704.05295** (2017). http://arxiv.org/abs/1704.05295.