Information Systems 80 (2019) 136–147

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Computing inter-document similarity with Context Semantic Analysis

Fabio Benedetti, Domenico Beneventano *, Sonia Bergamaschi, Giovanni Simonini

Department of Engineering "Enzo Ferrari", Università di Modena e Reggio Emilia, Italy

ARTICLE INFO

Article history:
Received 13 March 2017
Received in revised form 14 November 2017
Accepted 17 February 2018
Available online 19 February 2018

Keywords:
Knowledge base
Knowledge graph
Inter-document similarity
Similarity measures
Information Retrieval

ABSTRACT

We propose a novel knowledge-based technique for inter-document similarity computation, called *Context Semantic Analysis* (CSA). Several specialized approaches built on top of specific knowledge base (e.g. Wikipedia) exist in literature, but CSA differs from them because it is designed to be portable to any RDF knowledge base. In fact, our technique relies on a generic RDF knowledge base (e.g. DBpedia and Wikidata) to extract from it a *Semantic Context Vector*, a novel model for representing the context of a document, which is exploited by CSA to compute inter-document similarity effectively. Moreover, we show how CSA can be effectively applied in the Information Retrieval domain. Experimental results show that: *(i)* for the general task of inter-document similarity, CSA outperforms baselines built on top of traditional methods, and achieves a performance similar to the ones built on top of specific knowledge bases; *(ii)* for Information Retrieval tasks, enriching documents with *context* (i.e., employing the Semantic Context Vector model) improves the results quality of the state-of-the-art technique that employs such similar *semantic enrichment*.

1. Introduction

Recent years have seen a growing number of knowledge bases employed in several domains and applications. Besides DBpedia [1], which is the heart of the Linked Open Data (LOD) cloud [2], other important knowledge bases are: Wikidata [3], a collaborative knowledge base; YAGO [4], a huge semantic knowledge base derived from Wikipedia, WordNet and GeoNames; Snomed CT [5], the best known ontology in the medical domain and AGROVOC [6], a multilingual agricultural thesaurus we recently used for annotating agricultural resources [7].

In the literature, knowledge-based approaches have been employed for improving existing techniques in Natural Language Processing (NLP) [8] and Information Retrieval (IR) domains [9]. Yet, there is much room for improvement in order to effectively exploit these rich models in these fields [10]. For instance, in the context of inter-document similarity, which plays an important role in many NLP and IR applications, classic techniques rely solely on syntactic information and are usually based on Vector Space Models [11], where the documents are represented in a vector space having document words as dimensions. Nevertheless, such techniques fail in detecting relationships among concepts in simple scenarios like the following sentences: *"**The Rolling Stones** with the participation of **Roger Daltrey** opened the concerts' season in **Trafalgar Square**"* and *"The bands headed by **Mick Jagger** with the leader of **The Who** played in **London** last week"*. These two sentences contain highly related concepts (e.g., Roger Daltrey is the leader of The Who) which can be found by exploiting the knowledge network encoded within knowledge bases such as DBpedia.

To overcome the limitation of a purely syntactical approach, in [12] we proposed *Context Semantic Analysis* (CSA), a novel semantic technique for estimating inter-document similarity, leveraging the information contained in a knowledge base. One of the main novelties of CSA w.r.t. other knowledge-based approaches is its applicability over any RDF knowledge base, so that all datasets belonging to the LOD cloud [2] (more than one thousand) can be used. CSA is based on the notion of *contextual graph* of a document, i.e. a subgraph of the knowledge base that contains the contextual information of the document; the notion of *contextual graph* is very similar to the one of *semantic graph* defined in [10]. The contextual graph is then suitably weighted to capture the degree of associativity between its concepts, i.e., the degree of relevance of a property for the entities it connects. The vertices of such a weighted contextual graph are then ranked by using *PageRank* methods, so obtaining a *Semantic Context Vector*, a novel model able to represent the *context* of the document. Thus, the similarity of two documents is computed by comparing their Semantic Context Vectors with general vector comparison methods, such as the *cosine similarity*. By evaluating our method on a standard benchmark for document similarity (which consider correlations with human judges), we showed how CSA outperforms almost all other methods and how it can exploit any RDF knowledge

* Corresponding author.
E-mail addresses: Fabio.Benedetti@unimore.it (F. Benedetti), Domenico.Beneventano@unimore.it (D. Beneventano), Sonia.Bergamaschi@unimore.it (S. Bergamaschi), Giovanni.Simonini@unimore.it (G. Simonini).

https://doi.org/10.1016/j.is.2018.02.009
0306-4379/

base. Moreover we analyzed its scalability in a clustering task with a large corpus of documents, and showed that our approach outperforms the considered baselines.

This paper extends our previous work at the SISAP 2016 Conference. The main novel contribution of the extended paper is to test Context Semantic Analysis (CSA) applicability and effectiveness in a real-world application domain, such as Information Retrieval (IR). To this purpose, we analyzed the semantic based approaches recently proposed in the Information Retrieval research community. We found that, the most effective and general IR framework, adopting semantic enrichment of documents, is KE4IR [13]. We studied its layered architecture and tried to improve its performance, by including CSA, as a new semantic layer. The outcome was really positive as we were able to show that KE4IR + CSA outperforms the original KE4IR framework (see Section 5.2).

The paper is structured as follows. Section 2 describes the related work, while Section 3 is devoted to some preliminaries useful for the rest of the paper. Then, CSA is described in Section 4 and Section 5 shows its evaluation. Finally, Section 6 outlines conclusions and future work.

2. Related work

Text similarity has been one the main research area of the last years due to wide range of its applications in tasks such as information retrieval, text classification, document clustering, topic detection, etc. [14]. In this field a lot of techniques have been proposed but we can group them in two main categories, *content-based* and *knowledge-enriched* approaches, where the main difference is that the first group uses only textual information contained in documents while the second one enriches these documents by extracting information from other sources, usually knowledge bases.

2.1. Content-based approaches

The standard document representation technique is the *Vector Space Model* [11]. Each document is expressed as a weighted high-dimensional vector, the dimensions corresponding to individual features such as words. The result is called the *bag-of-words* model and it is the first example of *content-based* approach. The limitation of this model is that it does not address polysemy (the same word can have multiple meanings) and synonymy (two words can represent the same concept). Another technique belonging the *content-based* group is Latent Semantic Analysis (LSA) [15], which assumes that there is a latent semantic structure in the documents it analyzes. Its goal is to extract this latent semantic structure by applying dimensionality reduction to the terms-document matrix used for representing the corpus of documents.

Finally, in the context of Information Retrieval, *probabilistic models* are employed for ranking documents according to their relevance (similarity) to a given search query, i.e., similarities are computed as probabilities that a document representation matches or satisfies a query. Among them, the most popular are: Okapi BM-25 [16] and *language modeling* approaches [17].

2.2. Knowledge-enriched approaches

Recently, a lot of effort has been employed in designing new techniques for text similarity that use information contained in knowledge bases. Explicit Semantic Analysis (ESA) [18] proposes to map the documents to Wikipedia articles, and to represent each document as a vector of features extracted from both the document and the related articles text. Thus, the similarity of two documents can be computed through any vector space comparison algorithm.

Another document similarity technique that leverages the information contained in Wikipedia is WikiWalk [19], where the personalized PageRank on Wikipedia pages is used, with a personalization vector based on the ESA weights on concepts detected in the documents, to produce a vector used for estimating the similarity. A big drawback of this approach is the computational cost; indeed, for each document we have to execute first ESA and then compute the personalized PageRank on the whole Wikipedia. Another remarkable approach is SSA, i.e. Salient Semantic Analysis [20]. This method starts with Wikipedia for creating a corpus where concepts and saliency are explicitly annotated, then, the authors use this corpus to build concept-based word profiles, which are used to measure the semantic relatedness of words and texts. These groups of *knowledge-enriched* approach are designed for using only Wikipedia as source of knowledge and they are not portable to generic knowledge bases. Our method CSA differs from them because it aims to be a general approach that can use any knowledge base expressed according to the Semantic Web standard, i.e described in RDF, so that all datasets belonging to the Linked Open Data cloud [2] (more than one thousand) can be used as source of knowledge. To the best of our knowledge, the only approach portable to knowledge bases is the one proposed in [10], where the authors represent documents belonging to a corpus as graphs extracted form a RDF knowledge base. It differs from CSA because it is based on a Graph Edit Distance (GED) graph matching method to estimate similarity, while in our approach a document is represented as a vector and the similarity can be estimated more effortlessly by using cosine similarity.

Finally, from the Information Retrieval community, two recent works [13,21] have proposed general information retrieval techniques, based on the Vector Space Model, to work with documents semantically enriched with Linked Open Data. In Section 5.3, we show how CSA can be employed to enhance the IR framework KE4IR [13], which has been experimentally demonstrated to outperform Waitelonis et al. [21]. Our experimental evaluation shows that CSA improves original KE4IR.

3. Preliminaries

3.1. Inter-document similarity

The state-of-the-art techniques for estimating inter-document similarity are primarily based on *Vector Space Models*: a document is represented through a *bag-of-words* feature vector, which contains information about the presence and absence of words in the document, and the similarity between two documents is calculated as the cosine of the angle between the two respective vectors (i.e., their *cosine similarity*).

Vector Space Models are generally based on a co-occurrence matrix, a way of representing how often words co-occur; in a *term-document matrix*, each row represents a word and each column represents a document. Let C be a corpus composed of n documents, where each document d_j is composed of a sequence of terms. Let m be the number of terms in C; the *term-document matrix* T is a matrix $m \times n$ where each cell (i, j) contains the weight t_{ij} assigned to term i in the document j. A document d_j is then represented by the *vector* $\vec{d_j} = [t_{1j}, \ldots, t_{mj}]$.

While with the simple *bag-of-words* representation the weight t_{ij} is equal to the number of time the term i appears in the document j, many weighting strategies have been proposed in the literature (see, for example, [22]), such as *tf-idf* (*Term Frequency - Inverse Document Frequency*).

The novel technique we proposed *Context Semantic Analysis* (CSA), is based on a matrix T whose columns are associated with documents, and whose rows with concepts of a *Knowledge Base KB*, such as DBpedia (Section 3.2). The weight assigned to concept i in

document j intuitively defined as follows: first, the document j is represented by means of the so-called *Contextual Graph* (Section 4); the weight of the concept i into the document j is then computed as the relevance of the node i in the *Contextual Graph*, by using well-known algorithms, such as *PageRank* (Section 3.3).

The aim of the proposed technique is to extend documents with a *context* extracted from a knowledge base; to show that this extension is useful for estimating document similarity, we chose to use common approaches, such as *tf-idf* with the *cosine similarity*. The combination of our technique with more complex weighting schemes represent an interesting future work.

3.2. Knowledge base

We focus on RDF knowledge bases[1]; an RDF knowledge base can be considered a set of facts (statements), where each fact is a triple of the form $\langle subject, predicate, object \rangle$. A set of such triples is an *RDF graph* $KB = (V, E)$: a labeled, directed multi-graph, where subjects and objects are vertices and the predicates are labeled edges between vertices. According to [23], vertices are divided in 3 disjoint sets: URIs U, blank nodes B and literals L. Literals cannot be the subjects of RDF triples.

The triples of an RDF knowledge base can usually be divided into *A-Box* and *T-Box*; while the *A-Box* contains instance data (i.e. extensional knowledge), the *T-Box* contains the formal definition of the terminology (classes and properties) used in the *A-Box*. As an example, Fig. 1 shows an extract of DBpedia[2]; in the DBPedia T-Box, the property *dbo:genre* is defined with `rdfs:range` `dbo:MusicGenre` and the class `dbo:Band` is defined as a sub-class of both `dbo:Organization` and `dbo:Group`. In the DBPedia A-Box, the instance data `dbr:The_Rolling_Stones` (an instance of the class `dbo:Band`) is connected by the property `dbo:genre` to the instance data `dbr:Rock_music` (an instance of the class `dbo:MusicGenre`).

For our experiments we choose two generic domain knowledge bases: DBpedia [1] and Wikidata [3], due to their large coverage and variety of relationships at the extensional level.

3.3. PageRank

PageRank was first proposed to rank web pages [24], but the method is now used in several applications for finding vertices in a graph that are most relevant for a certain task. Let G be a graph with n vertices and d_i be the outdegree of the vertex i; the *Standard PageRank* algorithm computes the *PageRank vector* R defined by the equation:

$$R = cMR + (1 - c)v$$

where M is the *transition probability* matrix, a $n \times n$ matrix given by $M_{ij} = 1/d_i$ (d_i is the outdegree of i) if it exists an edge from i to j and 0 otherwise, c is the *damping factor*, a scalar value between 0 and 1 (usually between 0.85 to 0.95) and v is the *teleport vector*, a uniform vector of size n in which each element is $1/n$.

In the Standard PageRank configuration the vector v is a stochastic normalized vector where all the values are $\frac{1}{n}$, meaning that the random surfer has an equal probability to be teleported in any of the nodes of the graph G. In other words, Standard PageRank uses just graph topology; on the other hand, many graphs, as the ones in our case, come with weights on either nodes and edges, which can be used to *personalize* the PageRank algorithm. The *Personalized PageRank* [25] uses *node weights* to define a non-uniform vector v and thus biasing the computation of the *PageRank vector* R to be more influenced from heavier nodes. Another variant is the *Weighted PageRank* [26] which uses *edge weights* to define a custom transition probability matrix for influencing further the computation of the *PageRank vector* R. In the transition probability matrix of the *Weighted PageRank*, a *weighted outdegree* d_i for a node i is used, with $d_i = \sum_j A_{ij}$, where $A_{ij} > 0$ represents the weight on an edge from node i to node j.

4. Context semantic analysis

In this section we introduce our novel technique for estimating inter-document similarity, called *Context Semantic Analysis* (CSA), that is based on leveraging the information contained in a generic RDF knowledge base. Given a corpus C of documents and an RDF knowledge graph KB, CSA is composed of the following three steps:

1. **Contextual Graph Extraction**: the *Contextual Graph* $CG(d)$ containing the contextual information of a document d is extracted from the KB.
2. **Semantic Context Vectors Generation**: the *Semantic Context Vector* $SCV(d)$ representing the context of the document d is generated analyzing its $CG(d)$.
3. **Context Similarity Evaluation**: the *Context Similarity* is evaluated by comparing the context vectors of documents belonging to the corpus C.

4.1. Contextual graph extraction

Given a document d and a knowledge graph KB, the goal of this first step is to extract a subgraph from KB containing all the information about d. Our method relies only on the extensional knowledge of a knowledge base, i.e. on its *A-Box*. More precisely, given a knowledge base KB, we consider the subgraph $KB_A = (V_A, E_A)$ where the triples are in the *A-Box* of the KB. We also exclude the triples containing literals, so, all the vertices V_A belongs to $(U \cup B)$, i.e., are URIs or blank nodes, and every edge E_A corresponds to an *object property*. We made this choice because our previous works shown that the *T-Box* of several knowledge bases belonging to the LOD cloud is incomplete and sometimes even absent, moreover, information about the structure of a knowledge base can be inferred from its *A-Box* [27,28]. For example, in Fig. 1 we have only 3 triples that belongs to KB_A: the ones containing the `dbo:genre` property.

Given the subgraph KB_A, the extraction of the Contextual Graph $CG(d)$ for a document d is a three-step process:

1. Starting Entities Identification;
2. Contextual Graph Construction;
3. Contextual Graph weighting.

Such steps are described below.

1. Starting Entities Identification: the entities of KB_A explicitly mentioned in the document d are identified. Such set of entities is called *starting entities* of d, denoted by $SE(d)$. The problem of finding the set $SE(d)$ is an instance of the well-known *Named Entity Recognition* problem [29]. Its solution is out of scope of this work, thus we empirically evaluated some of the already implemented techniques, and, on the basis of the obtained results, we chose DBpedia Spotlight [30] and TextRazor[3] to identify starting entities w.r.t. DBpedia and Wikidata, respectively.

2. Contextual Graph Construction: the Contextual Graph of the document d is defined as the subgraph of KB_A composed of all the

[1] https://www.w3.org/TR/rdf-primer/.

[2] We abbreviate URI namespaces with common prefixes, such as `dbpedia.org/resource/`(prefix `dbr`), `dbpedia.org/property`(prefix `dbp`) and `dbpedia.org/ontology`(prefix `dbo`); see http://prefix.cc for details.

[3] https://www.textrazor.com/.

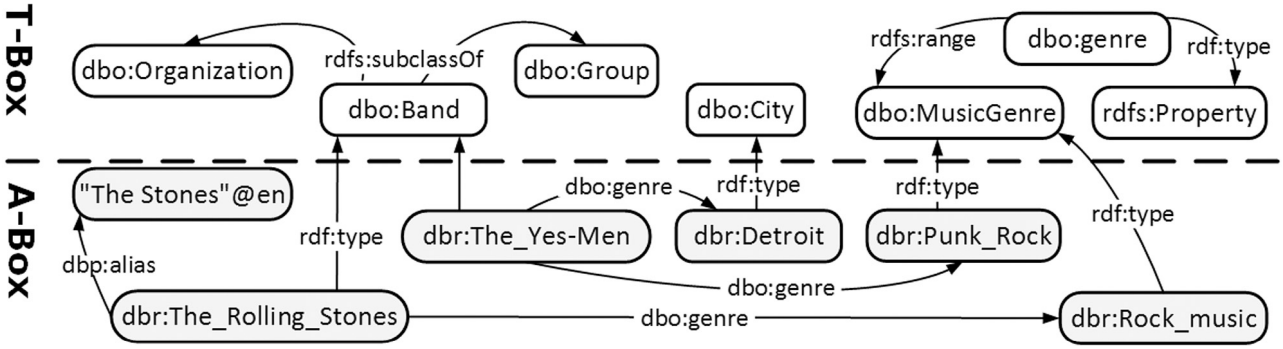

Fig. 1. Example of an RDF KB, with the *A-Box* and the *T-Box*.

triples connecting with a path of length l, at least 2 starting entities in $SE(d)$. More precisely, given a document d and a length $l > 0$, we define:

$$CG_l(d) = \{\langle s, p, o\rangle | \langle s, p, o\rangle \in KB_A \wedge \langle s, p, o\rangle \in Path(s_1, s_2) \wedge length(Path(s_1, s_2)) \leq l \wedge s_1, s_2 \in SE(d)\}$$

where $Path(s_1, s_2)$ is a path on KB_A from s_1 and s_2.

For example, let us consider the two sentences used in the introduction (each sentence is represented as a document):

d_1 : *"**The Rolling Stones** with the participation of **Roger Daltrey** opened the concerts' season in **Trafalgar Square**".*

d_2 : *"The bands headed by **Mick Jagger** with the leader of **The Who** played in **London** last week".*

The related starting entities in DBpedia are the following:

$SE(d_1) =$ { `The_Rolling_Stones`, `Roger_Daltrey`, `Trafalgar_Square`}

$SE(d_2) =$ {`Mick_Jagger`, `The_Who`, `London`}

In this example, by using l=2 we obtain $CG_2(d_1)$ with 5 nodes and $CG_2(d_2)$ with 12 nodes; by using l=3 we obtain $CG_3(d_1)$ with 141 nodes and $CG_3(d_2)$ with 66 nodes. The most significant portion of information shared between $CG_3(d_1)$ and $CG_3(d_2)$ is shown Fig. 2; in $CG_3(d_2)$ there is a path of length 1 between `London` and `Mick_Jagger`, while `Mick_Jagger` and `The_Who` are connected by means of two (different) paths, both of length 3.

In Information Retrieval, a keyword query is usually composed of a few words, so, in this context for a generic query q it is common to have only a single starting entity (i.e., $|SE(d_1)|$=1). A user, in order to retrieve the documents d_1 or d_2 could use keywords queries like:

q_1 : **Roger Daltrey**.

q_2 : **Mick Jagger**.

In Fig. 3 a portion of the Contextual Graphs extracted starting from these two queries are shown. The contextual graph $CG_3(q_1)$ contains, besides `dbr:Roger_Daltrey`, the entities `dbr:The_Who` and `dbr:Rock_music`, which belong to the contextual graph $CG_3(d_2)$ as well. Then, the query q_1 can retrieve both documents d_1 and d_2. Similar considerations can be done for contextual graph $CG_3(q_2)$ of the query q_2.

3. Contextual Graph weighting: In the literature, several graph weighting methods have been proposed to capture the degree of associativity between concepts in the graph, i.e., the degree of relevance of a property for the entities it connects [10,31]. The most common way of weighing a property p_i is to compute its *Information Content (IC)*, $IC(X = p_i) = -log(P(p_i))$, where $P(p_i)$ is the probability that a random variable X exhibits the outcome p_i. This metric makes the hypothesis that specificity is a good proxy for relevance; in our example, an edge labeled with `rdf:type` will accordingly get an *IC* which is comparably lower than, say, one labeled with `dbo:genre`. The metric $IC(p_i)$ measures the specificity of the property p_i, regardless of the entities it actually connects; to take into account that the same property can connect more or less specific entities, the authors in [10] considered $IC(obj_i|p_i)$ computed in a similar way to $IC(p_i)$, where $P(obj_i|p_i)$ is the conditional probability that a node obj_i appears as object of the property p_i; then they proposed the *Joint Information Content* weighting function: $w_{jointIC} = IC(obj_i|p_i) + IC(p_i)$. In our example, with this metric, the `rdf:type` edge leading to `dbo:MusicGenre` receives a much higher weight than that pointing to the far more generic `dbo:City`. The drawback of this function is that it penalizes infrequent object that occur with infrequent properties; for example, `dbo:Punk_Rock` is overall very infrequent, but it get an high probability when it occurs conditional on `dbo:genre`. The authors in [10] propose to mitigate this problem by computing the joint information content while making an independence assumption between the predicated and the object; the resulting weights are then computed as the sum of the Information Content of the predicate and the object, so obtaining the *Combined Information Content* $w_{combIC} = IC(obj_i) + IC(p_i)$.

The metrics presented so far take into account only the extensional knowledge of a *KB*, i.e. only on the triples of the *A-Box*; we introduce a new weighting function based on the fact that the importance of a property between two entities also depends on the classes to which such entities belong (each entity in an RDF graph is instance of at least one class). For example, in Fig. 1, most people would agree that, for subjects which are instance of `dbo:Band`, the importance of `dbo:genre` increases when the object is an instance of `dbo:MusicGenre`. In fact, the 94% of the `dbo:Band` instances are subject of a `dbo:genre` property that has as object, in 91% of cases, an instance of `dbo:MusicGenre`, while only the 0.002% of times, an instance of `dbo:City`. Taking in exam the triple $\langle s_i, p_i, o_i\rangle$, we measure the correlation between a property p_i, the class of the subject s_i and the class of the object o_i by using the notion of *Total Correlation* [32], which is a method for weighting multi-way co-occurrences according to their importance:

$$TotalCorrelation(s_i, p_i, o_i) = -log(\frac{P(S_i, p_i, O_i)}{P(S_i)P(p_i)P(O_i)})$$

where S_i and O_i are the classes associated to the entities s_i and o_i, respectively.[4]

[4] When an entity is an instance of more than one class we use the class with the minor number of instances because it better characterizes an entity; however if we filter the knowledge bases by excluding classes defined in external sources such as YAGO, GroNames, etc. only 6.4% of entities in Dbpedia and 2.22% in Wikidata are instances of more than one class.

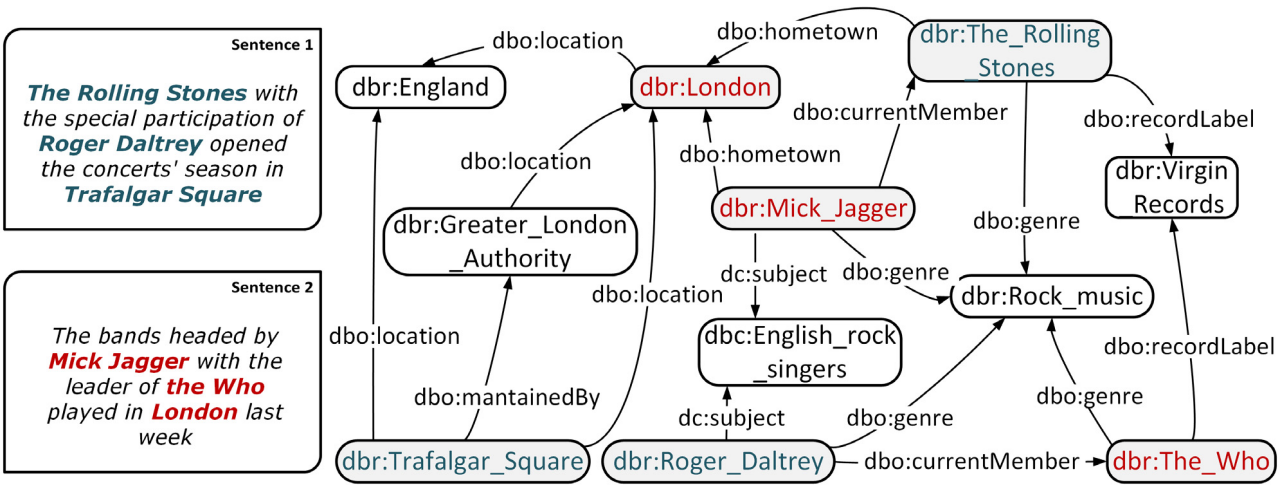

Fig. 2. Portion of DBpedia containing the most significant shared contextual information between the two sentences on the left.

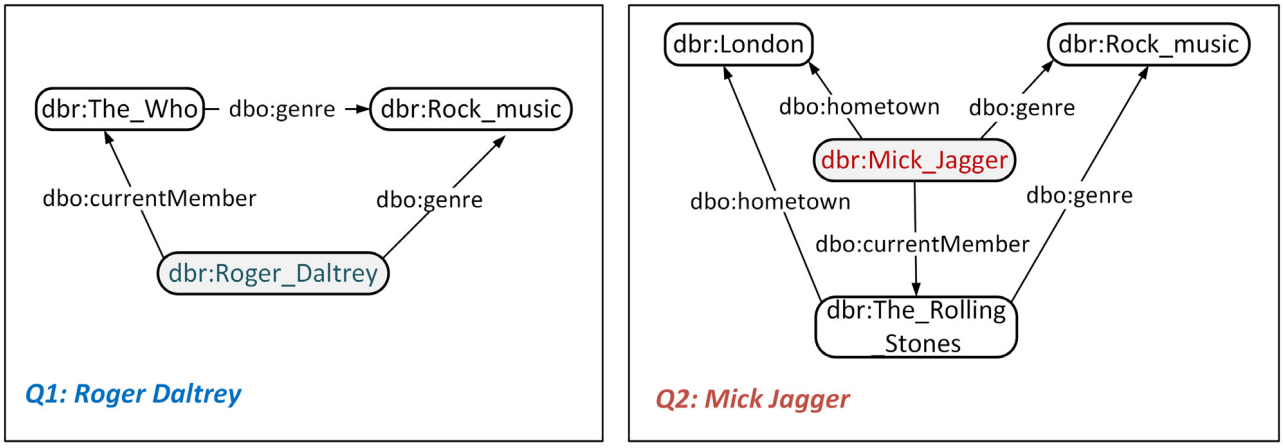

Fig. 3. A portion of the Contextual Graph extracted from DBpedia for the two keyword queries: *Roger Daltrey* and *Mick Jagger*.

To summarize, for Contextual Graphs we will consider three edge weight functions: Joint Information Content (W_{Joint}), Combined Information Content (W_{Comb}) and Total Correlation (W_{TotCor}).

4.2. Semantic context vectors generation

At this point we have all the ingredients necessary to define the notion of *Semantic Context Vector*, a vector representation of documents based on Contextual Graphs. Given a corpus of documents $C = \{d_1, \ldots, d_n\}$ and an RDF *KB*, for each document $d \in C$ we build its contextual graph $CG_l(d)$; then we consider the set $E = \{e_1, \ldots, e_m\}$ of entities occurring in all the contextual graphs. Similar to the *term-document matrix* (see Section 3.1) we consider an *entity-document matrix* T, a $m \times n$ matrix where the cell (i, j) contains the weight $s(e_i, d_j)$ of the entity $e_i \in E$ in the document $d_j \in C$. A document d_j is thus represented by the *jth* column of such matrix, called *Semantic Context Vector* of d_j and denoted by $SCV(d_j)$:

$$SCV(d_j) = (s(e_1, d_j), \ldots, s(e_m, d_j))$$

The weight $s(e_i, d_j)$ has to take into account for the importance of the entity e_i within $CG_l(d_j)$ and, thus, it is defined by considering an edge weight function and a PageRank method.

As edge weight functions for $CG_l(d)$, we consider W_{Comb}, W_{Joint} and W_{TotCor} (defined in the previous section) to set up the transition probability matrix M as a $k \times k$ matrix, where k is the number of nodes of $CG(d)$ and $M_{pq} = \frac{w(p,q)}{\sum_{z=1}^{k} w(p,z)}$, where $w(p, q)$ returns the weight if an edge from p to q exists, otherwise it returns 0. Moreover, we denote with $W_{noweight}$ the case when edge weights are not used and the transition probability matrix M is given by $M_{pq} = 1/d_p$ if it exists an edge from p to q and 0 otherwise (d_p be the outdegree of the vertex p).

The PageRank methods we consider are the ones resumed in Section 3.3:

1. *Standard PageRank*: in this case (denoted by r) there is no personalization vector, i.e., an uniform vector is considered;
2. *Personalized PageRank*: in this case (denoted by pr) the personalization vector $\vec{v} = (v_1, \ldots, v_k)$ is setup to give an equal probability to starting entities: $v_i = 1/|SE(d)|$ if $e_i \in SE$ and 0 otherwise.

As *damping factor* we consider a range of values from 0.10 to 0.95 with a step of 0.05.

To summarize, the *Semantic Context Vector* of a document d, $SCV(d)$, is defined by the following four parameters:

1. *KB* : the RDF Knowledge Base used to build the contextual graph $CG_l(d)$ of d;
   we used $KB = Dbpedia$ and $KB = Wikidata$ in our tests.
2. *CG-L* : the length for the Contextual Graph $CG_l(d)$;
   we tested our method with $CG\text{-}L = 2$ and $CG\text{-}L = 3$.
3. *WF*: the edge weight function for $CG_l(d)$:
   we consider W_{Comb}, W_{Joint}, W_{TotCor} and $W_{noweight}$.

Table 1
Semantic Context Vectors of the two documents of Fig. 2.

Entity	Document d_1		Document d_2	
	pr@75	r@75	pr@75	r@75
The Rolling Stones	**.187**	**.036**	.098	.082
Roger Daltrey	**.140**	**.018**	–	–
Trafalgar Square	**.155**	**.024**	–	–
London	.111	.048	**.225**	**.072**
Mick Jagger	.000	.024	**.155**	**.051**
The Who	.055	.028	**.175**	**.053**
England	.083	.050	.104	.090
Rock music	.072	.037	.098	.077

Table 2
PageRank and Personalized PageRank configurations.

Name		Dumping Factor
PageRank	Personalized PageRank	
r@50	pr@50	0.50
r@75	pr@75	0.75
r@85	pr@85	0.85
r@90	pr@90	0.90
r@95	pr@95	0.95

4. *PageRankConfiguration*: the *damping factor* and personalization vector used.
   With *r@df* and *pr@df* we denote Standard and Personalized PageRank, respectively, with a damping factor equal to *df*.

As an example, for the documents d_1 and d_2 of Fig. 2, part of their *SCVs* are shown in Table 1; the knowledge base is DBpedia and *CG-L* is equal to 3; both PageRank and Personalized PageRank are considered, with a damping factor equal to .75 (i.e. *r@75* and *pr@75*).

We can observe that PageRank tends to arrange weight in all the context graph's nodes, while with the Personalized PageRank all the weight is focused in the neighborhood of the starting entities.

Table 2 shows the different configuration used.

4.3. Context similarity evaluation

In this last step, the *Context Similarity* between two documents is evaluated by comparing their context vectors. More precisely, the *CSA Similarity*, denoted by sim_{CSA}, between two documents d_1 and d_2 is computed as the *cosine similarity* between their Semantic Contextual Vectors:

$$sim_{CSA}(d_1, d_2) = \frac{v \cdot s}{|v| \cdot |s|} = \frac{\sum_{i=1}^{n} v_i \cdot s_i}{\sqrt{\sum_{i=1}^{n} v_i^2} \cdot \sqrt{\sum_{i=1}^{n} s_i^2}}$$

where v=$SCV(d_1)$ and s=$SCV(d_2)$.

As an example, by considering the Semantic Context Vectors shown in Table 1, the sim_{CSA} between the two documents d_1 and d_2 of Fig. 2, is equal to 0.78 by using *r*@75 vectors and 0.61 by using *pr*@75. In the next section we will evaluate which CSA configuration is more effective in detecting similarities between documents.

4.3.1. Linear combination of CSA with text similarity measures

The CSA similarity, sim_{CSA}, is only based on information extracted from a knowledge base; to include in the final similarity measure (sim_f) also the textual information, we consider a linear combination of the CSA similarity with (standard) textual similarity measures sim_{TXT} (such as LSA [33] and ESA [34]) between two documents as:

$$sim_f(d_1, d_2) = \alpha \cdot sim_{CSA}(d_1, d_2) + (1-\alpha) \cdot sim_{TXT}(d_1, d_2)$$

where α is the weight parameter used for combining the two measures.

5. Evaluation

In this section we evaluate CSA: firstly, we assess CSA efficacy by considering the correlation with human judges; secondly, we evaluate how CSA performs in a real-world application, employing it in an Information Retrieval framework; thirdly, we analyze CSA scalability in a clustering task on a large dataset.

All experiments have been performed on a server running Ubuntu 14.04, with 80 GB RAM, and an Intel Xeon E5-2670 v2 @ 2.50 GHz CPU. CSA has been implemented in Python 2.7, and for generating the contextual graphs, we imported the DBpedia graph in Neo4J.[5]

5.1. Correlation with human judges

This experiment compares on a benchmark dataset [33] results obtained with CSA, and results produced by human judgment.

5.1.1. Experimental setup

The most common and effective way for evaluating techniques of inter-document similarity is to assess how the similarity measure produced emulates human judges. To this end, we use the dataset of documents LP50[6] [33], which contains 50 documents, selected from the Australian Broadcasting Corporation's news mail service, evaluated by 83 students of the University of Adelaide. Each possible pair of documents (1225 pairs in total) has 8–12 human judgments.These judgments have been averaged for each document pair, obtaining only 67 distinct values for 1225 similarity scores. For this reason, Gabrilovich et al. [18] and Schuhmacher et al. [10] suggest to employ Pearson's linear correlation coefficient (r) between the computed similarities and the ones assigned by human judges. We follow this suggestion, to compare our results with those presented in [18] and [10]; yet, we also consider the Kendall's (τ) correlation coefficient, which is typically employed in Information Retrieval context to measure ordinal associations. As shown in the following, the outcome of our analysis shows that these two measures leads to the same conclusions for this experiment.

5.1.2. Results and discussion

In Table 3, CSA[7] is compared with other literature techniques. Bag-of-Words [33] indicates the simple bag-of-word document representation, coupled with *term-frequency* weighting and cosine similarity. We considered also Okapi BM25[8] as weighting, coupled with the dot product. Un-Backgrounded LSA means that LSA [33] has been applied only considering the LP50 dataset, differently form Backgrounded LSA, which employs additional documents to perform a better dimensionality reduction (see [33] for the details).

5 https://neo4j.com.

6 https://webfiles.uci.edu/mdlee/LeePincombeWelsh.zip.

7 Employing Total Correlation and *pr*@70, which has been shown to be the best setting for all the experiments we executed.

8 https://github.com/RaRe-Technologies/gensim.

Table 3
System comparison on the LP50 dataset.

	Pearson coefficient (r)	Kendall coefficient (τ)
CSA	0.62	0.35
CSA + LSA	0.65	0.39
CSA + ESA	0.72	0.42
Bag-of-Words [33]	0.41	0.13
BM25	0.50	0.17
Un-Backgrounded LSA [33]	0.52	0.18
Backgrounded LSA [33]	0.59	0.28
ESA reimplemented [34]	0.59	0.30
GED-based (Dbpedia) [10]	0.63	0.37
SSA [20]	0.68	0.40
WikiWalk + ESA [19]	0.77	0.47

The original performance of ESA reported in [18] on the LP50 dataset has been criticized in [34] for being based on a cut-of value used to prune the vectors in order to produce better results on the LP50 dataset and, consequently, over-fit the approach to this particular dataset. In fact, a much lower performance has been obtained in [34] and [20] by re-implementing ESA without adapting the cut-off value. We employ this implementation in our experiments.

The main result emerging from this comparison is that our CSA method alone yields results comparable to state-of-the-art techniques (LSA and ESA), and enhances them when used in conjunction; for example, CSA + ESA obtains a correlation $r = 0.72$ ($\tau = 0.42$), so it attains a 16% improvement. The Graph Edit Distance (GED) based approach of [10], which is the most similar to our, produces almost identical results but with GED the similarity measures are obtained in a much more computationally expensive way than in CSA (a deeper comparison is in the next Section). By taking in exam other *knowledge-enriched* techniques built on top of a specific knowledge base (Wikipedia), CSA combined with ESA slightly outperforms SSA, but it does not reach the performance of WikiWalk + ESA.

As shown in Table 3, the relative performances of the methods are the same either considering the Pearson's r and the Kendall's τ. In fact, we observe that these two measures show the same trends in all our experiments; hence, hereafter we present only the results for Pearson's r for the sake of presentation.

Complete results are shown in Fig. 4, which shows the Pearson coefficient r between the human gold standard and CSA by varying the parameters that define the Semantic Context Vectors, with the exception of *CG-L* that has been considered constant and equal to 3. One of the main results is that, for all the configurations, the Personalized PageRank (pr) outperforms the Standard PageRank (r); another interesting result is that, in almost all the configurations, the novel edge weighting function W_{TotCor} we proposed slightly outperforms the other ones, W_{Joint} and W_{Comb}. We can also appreciate different behaviors w.r.t the KB: DBpedia is more stable, while Wikidata exhibits a strong performance decay by increasing the damping factor, with the Personalized PageRank. In particular, the CSA configuration with DBpedia, W_{TotCor}, Personalized PageRank with damping factor ranging from 0.30 to 0.85, is quite stable: it varies by only 2.5% from the minimum (0.605 pr@30) to the maximum (0.62 pr@65); then such a CSA configuration is almost parameter free.

Table 4 shows the Pearson coefficient r for the best CSA configurations we found, by varying all the parameters.

In order to evaluate CSA we produced some baselines:

- Jaccard on *starting entities*: we used the *starting entities* collected for each document as descriptor of the document and we used the Jaccard similarity for estimating the similarity between documents, namely $sim(d_1, d2) = \frac{SE(d_1) \cap SE(d_2)}{SE(d_1) \cup SE(d_2)}$.
- Cosine (bag of words): we model the document corpus in a standard bag of words Vector Space Model and we compute the cosine similarity.[9]

CSA is able to outperform both baselines; we obtained a relative improvement of the 21% (with either DBpedia and Wikidata) w.r.t. the Jaccard baseline[10]; this improvement is particularly significant because it is only due to the information extracted from the knowledge bases by CSA.[11] W.r.t. the Cosine baseline the margins are greater (34% DBpedia and 33% Wikidata); this result is not too surprising because this baseline utilize only the words contained in the text for estimating the similarity.

Table 5 shows the performance of the linear combination of CSA with the standard text similarity measures un-backgrounded LSA [33][12] and ESA reimplemented [34]. The best performance is obtained with $\alpha = 0.5$, and we can observe that the best configurations obtained in Table 4 for CSA (i.e. *pr@65* for DBpedia and *pr@40* for Wikidata) are also the best configurations of CSA combined with LSA and ESA.

5.2. Information retrieval application

The goal of this experiment is to evaluate CSA in a real-world Information Retrieval (IR) application. In particular, we integrated CSA in a IR framework (KE4IR [13]), and measure on a well-known benchmark dataset the improvement yielded by our technique.

5.2.1. Experimental setup

Given a text query, the goal of IR is to find the relevant documents in a text collection, ranking them according to their *relevance degree* for the query. The *relevancy* of documents is typically measured by means of similarity measure in the Vector Space Model, hence employing CSA for this task is straightforward. We consider KE4IR [13], based on the popular IR framework Apache Lucene.[13] To the best of our knowledge, KE4IR is the current state-of-the-art in IR for retrieving documents with semantic enrichment[14] (i.e., documents enriched with annotation derived from external knowledge bases, such as DBpedia).

In KE4IR both the documents and the queries are represented as *term vectors* whose elements are the weights of textual and

9 Implemented as in [33] (only removing the stopwords).

10 If not explicitly stated all the difference in performance are statistically significant at $p\text{-}value < 0.05$ using Fisher's Z-value transformation.

11 The sets of starting entities are obtained by using NER APIs.

12 With *tf-idf* as weighting function.

13 http://lucene.apache.org/.

14 Please refer to the original paper of Corcoglioniti et al. [13], where an extensive evaluation is performed, comparing KE4ER with other existing approaches, showing its superiority.

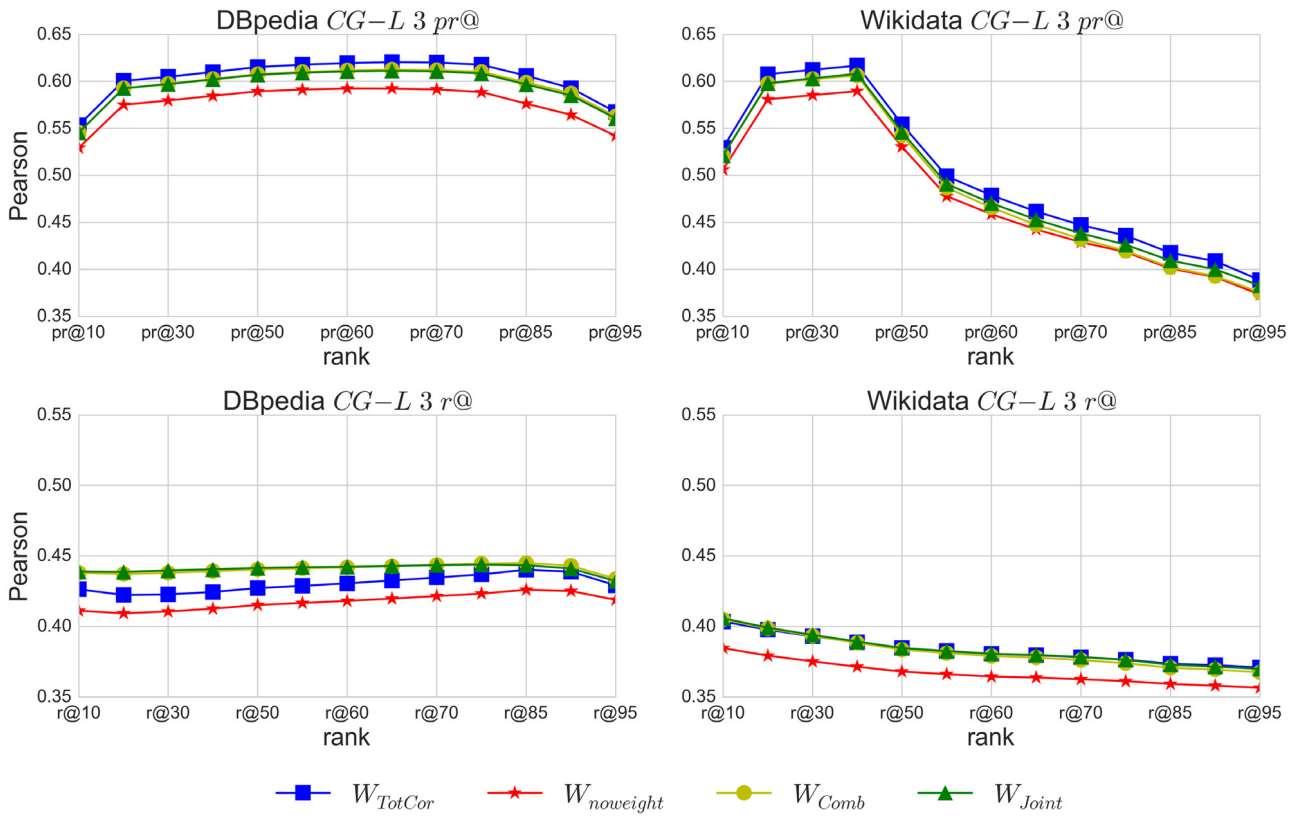

Fig. 4. Pearson correlation with human judgments (LP50 Dataset) of CSA, with different configurations.

Table 4
Results on the LP50 dataset (Pearson r correlation coefficient).

			$W_{noweight}$		W_{Comb}		W_{Joint}		W_{TotCor}		Best
DBpedia	*CG-L*	2	*pr*@40	0.57	*pr*@40	**0.59**	*pr*@60	0.58	*pr*@30	**0.59**	0.59
		3	*pr*@60	*0.59*	*pr*@65	0.61	*pr*@65	0.61	*pr*@65	**0.62**	0.62
	Jaccard on *starting entities*										0.49
			$W_{noweight}$		W_{Comb}		W_{Joint}		W_{TotCor}		
Wikidata	*CG-L*	2	*pr*@40	0.54	*pr*@40	0.56	*pr*@40	0.55	*pr*@40	**0.57**	0.57
		3	*pr*@40	0.59	*pr*@40	0.60	*pr*@40	0.60	*pr*@40	**0.61**	0.61
	Jaccard on *starting entities*										0.48
	Cosine (bag of words)										0.41

Table 5
Best Pearson correlation obtained on the LP50 dataset by combining CSA ($l = 3$ and *Total Correlation* as weight function) with LSA and ESA.

		Alpha value α					
		0.25		0.5		0.75	
DBpedia	CSA + LSA	*pr*@70	0.39	*pr*@70	0.39	*pr*@70	0.37
	CSA + ESA	*pr*@80	0.41	*pr*@65	**0.41**	*pr*@65	0.37
Wikidata	CSA + LSA	*pr*@40	0.38	*pr*@40	0.38	*pr*@40	0.36
	CSA + ESA	*pr*@40	0.41	*pr*@40	**0.42**	*pr*@40	0.39

semantic content extracted from DBpedia.[15] The terms derived directly from the text represent the *textual*-layer. The authors in [13] enriched the textual information with other layers, which are: the *uri*-layer, the *type*-layer, the *time*-layer, and the *frame*-layer.

- The *uri*-layer contains the entities of DBpedia related to the document/query text (e.g., `dbr:The_Rolling_Stones`) weighted according to their tf-idf of the entities in the documents. KE4IR employs PIKES[16] to annotate and enrich documents/queries.
- The *type*-layer is composed of the classes of the entities identified (e.g., `dbo:Band`).
- The *time*-layer contains the temporal values expressed in the text end matched against DBpedia (e.g., Year, Month, etc.).

[15] The weight of a term is its *tf-idf* computed as the frequency of the term in the document/query, multiplied by the inverse of the frequency of the term in the document collection. Notice that if a query term (textual or semantic) is not associated to any of the documents in the collection, then it does not contribute to the IR process, having no match in the document index.

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

- The *frame*-layer is composed of compact structures capturing relations among entities.

In order to compute the rank for each document d_i given a query q_j a similarity score for each of the layer is computed by using a measure $sim_{dot}(d, q)$ derived from the *cosine* similarity:

$$sim_{dot} = d \cdot q = \sum_{i=1}^{n} d_i \cdot q_i$$

Then, the similarity scores obtained for each of the layers are linearly combined to produce the final rank. Notice that dividing $sim_{dot}(d, q)$ for the product of the norms of the two vector d and q we obtain the cosine similarity of the two vectors. Omitting these normalization components is a common practice in the context of IR: this allows to avoid biased results due to the typically small size of the query terms [13].

We extended KE4IR to support CSA as independent layer, to employ it as substitute of the *uri*-layer in our experiments. (Notice that both layers are composed of entities extracted from a knowledge base used to represent the content of a document.) To compare *standard* KE4IR (i.e., KE4IR with the original *uri* layer) and KE4IR with the CSA-layer we employ the same dataset, the same experimental setup, and metrics of Corcoglioniti et al. [13].[17]

For this evaluation, we employed two **datasets** described in the following:

1. `yovisto`, which consists of a set of 331 documents from the *yovisto* blog[18] on history in science, tech, and art. The articles have an average length of 570 words, containing 3 to 255 annotations (average 83). Moreover, for this dataset the gold standard for the annotation is known, since documents have been manually annotated with DBpedia entities. Hence, employing this dataset, the performance of CSA can be measured minimizing the error introduced by *automatic named entity recognition tools*, such as DBpedia Spotlight [30] (which is employed only to spot entities in the queries, as described later).
2. `trec2001` [35], which is composed of $\sim 1.5 * 10^6$ documents extracted from the web.[19] No gold standard is provided for the annotation of this dataset, thus we employed DBpedia Spotlight to annotate these documents, as pre-processing step.

The generation of the contextual graphs ($CG_3(d)$ for each document d in the datasets) for the `yovisto` and `trec2001` dataset took $\sim$1 minute and $\sim$6 days, respectively. This time is required only once for dataset (as pre-processing) and could be significantly reduced employing data intensive scalable computing systems, such as Map Reduce and Apache Spark. Moreover, note that the contextual graphs computation can be incremental, thus when a new document is added to a collection, only its contextual graph has to be computed.

For the queries, `yovisto` and `trec2001`, provides 35 and 50 queries respectively, for which the list of relevance judgments is available. We limit our evaluation to the subset of 25 queries on `yovisto`, and 44 queries on `trec2001`, for which DBpedia Spotlight can spot entities. Notice that the limitation does not depend on CSA inherently, but rather on the coverage of entities contained in the knowledge base (DBpedia). Moreover, on `yovisto`, none of the queries contain more than one spotted entity, making this the ideal scenario for testing how CSA behaves with limited *context*, i.e., when $|SE| = 1$. (On `trec2001` only four queries contains more than one spotted entity).

For both documents and queries we extracted their $CG_3(d/q)$ using DBpedia as knowledge base and we computed their $SCV(d/q)$ for several configurations; then, we stored the *SCVs* for being used in the KE4IR framework. The metrics employed for measuring the performance are the Normalized Discounted Cumulative Gain (*NDCG*), Mean Average Precision (*MAP*), and Mean Reciprocal Rank (*MRR*), typically used to evaluate IR systems [36]:

- ***NDCG* assesses the overall ranking quality**.
  It takes into account both the relevance of a retrieved document and its position (notice that the relevance of a document is known from the available judgment employed as ground truth). It assumes values in the interval [0.0, 1.0], where 1.0 correspond to the maximum value obtained when all the relevant documents are retrieved, and their order matches the best ordering possible in terms of *relevance*.
- ***MAP* assesses the overall *precision quality***.
  It is obtained by averaging the precision measured after that each relevant document has been retrieved.[20] It assumes values in the interval [0.0, 1.0], where 1.0 is the best value. In contrast to NDCG, it does not take into account false negative; for instance, if only one document is retrieved and it is relevant the precision is 1, even when many more relevant documents exist.
- ***MRR* assesses the ranking quality of the *first correct result* retrieved**.
  It is computes averaging the *reciprocal ranks* over all the queries; where the reciprocal rank (RR) is the reciprocal of the highest ranking position of a correct answer given a query. It assumes values in the interval [0.0, 1.0], where 1.0 is the best value.

For IR approaches, it is common to assume that users only look at the "first page" of the results; hence, we record *NDCG* and *MAP* both for the complete result set, and for the top ten results (denoted by *NDCG@10* and *MAP@10*).

5.2.2. Results and discussion

The results of our experiment is summarized in Table 6. As far the exploitation of entity information is concerned, we observe that KE4IR with the CSA-layer (KE4IR w/ CSA) outperforms KE4IR (KE4IR w/ uri), both on `yovisto` and `trec2001` datasets (Table 6a-b). Differently, considering both the contribution of the information about the text and about the entities, the advantage of CSA is less evident, but still relevant: on `yovisto` (Table 6a), the employment of CSA allows to reach the highest performances, with the only exception for the MRR metric; while on `trec2001` (Table 6b), CSA wins for all the metrics. This confirms the results obtained by Corcoglioniti et al. [13]: the textual-layer represents the most important contribution to the final results. In fact, notice that textual-layer alone achieves higher results than the uri/CSA-layer alone. The improvements of KE4IR with the CSA-layer (KE4IR w/ CSA) over traditional KE4IR (KE4IR w/ uri) resulted statistically significant, according the paired t-test with a threshold p-value equal to 0.05.

As far as the query time performance is concerned, we did not record any significant difference in the time execution when employing KE4IR w/ CSA and KE4IR w/ uri.

Fig. 5 shows the performance metrics described above (*NDCG* and *NDCG@10*, *MAP* and *MAP@10*, *MRR*) on `yovisto` by varying two of the parameters that define the Semantic Context Vectors, i.e., the Contextual Graph weighting function and the dumping

17 We employed the original authors' implementation of KE4IR available at http://pikes.fbk.eu/ke4ir.html.

18 http://blog.yovisto.com/.

19 http://ir.dcs.gla.ac.uk/test_collections/wt10g.html.

20 The precision is defined as: $\frac{|\{RelevantDocuments\} \cap \{RetrievedDocuments\}|}{|\{RetrievedDocument\}|}$.

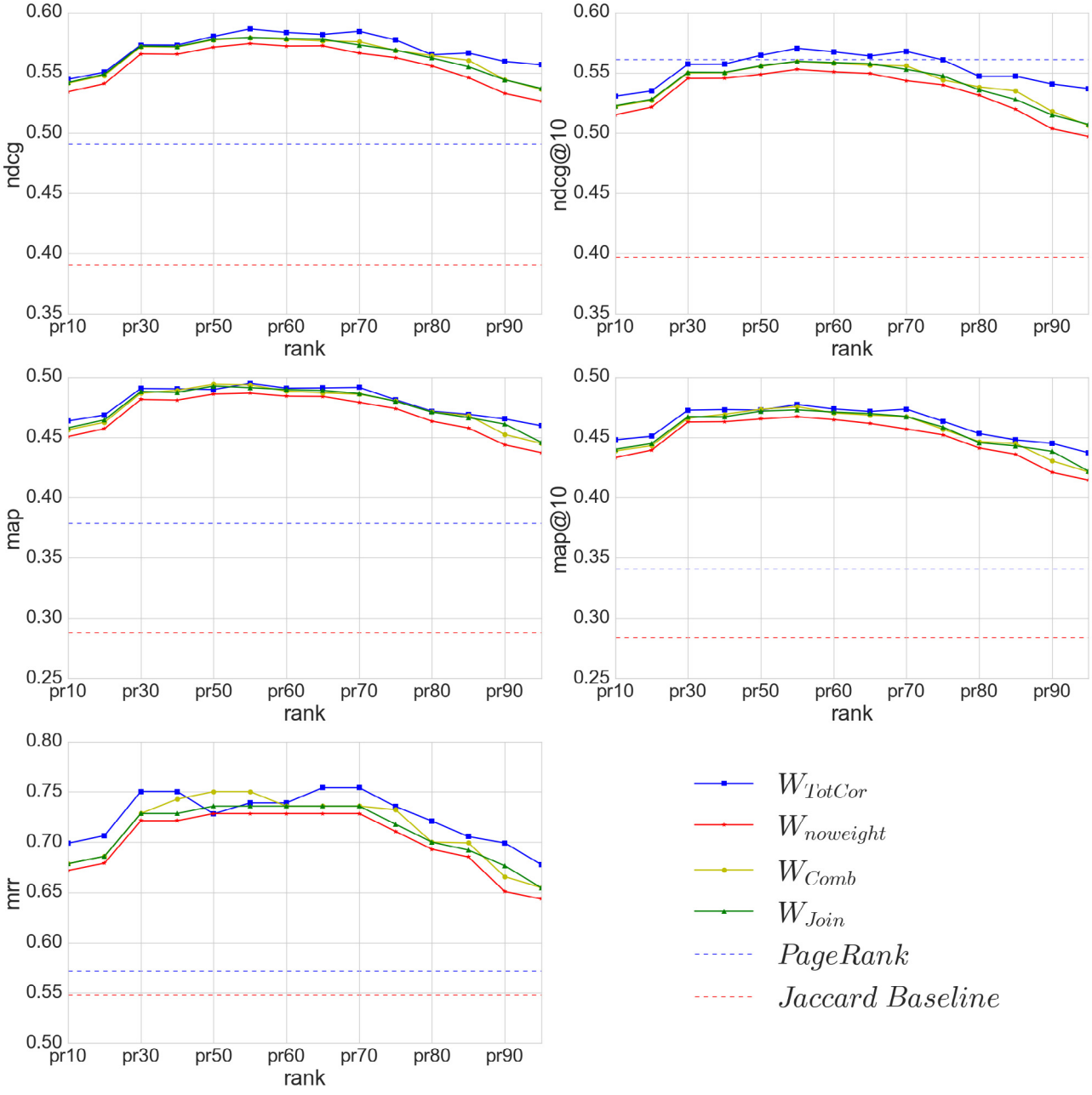

Fig. 5. CSA evaluation for the IR metrics on `yovisto` *NDCG* and *NDCG@10*, *MAP* and *MAP@10*, *MRR*.

Table 6
KE4IR experimental results.

Approach	Inf. employed	NDCG	NDCG@10	MAP	MAP@10	MRR
(a) `yovisto`						
KE4IR *w/* uri	Entities	0.461	0.431	0.334	0.310	0.580
KE4IR *w/* **CSA**	Entities	0.666	0.640	0.568	0.541	0.822
KE4IR *w/* textual	Text	0.756	0.681	0.596	0.546	0.907
KE4IR *w/* textual, uri	Text, Entities	0.864	0.818	0.751	0.710	**0.980**
KE4IR *w/* textual, **CSA**	Text, Entities	**0.898**	**0.853**	**0.784**	**0.746**	0.969
(b) `trec2001`						
KE4IR *w/* uri	Entities	0.109	0.090	0.069	0.049	0.019
KE4IR *w/* **CSA**	Entities	0.123	0.099	0.072	0.051	0.020
KE4IR *w/* textual	Text	0.499	0.336	0.289	0.122	0.053
KE4IR *w/* textual, uri	Text, Entities	0.478	0.351	0.286	0.137	0.054
KE4IR *w/* textual, **CSA**	Text, Entities	**0.501**	**0.375**	**0.292**	**0.153**	**0.058**

Table 7
Results on the Reuters 21578 (*re0*) dataset (F-measure and execution time for building the cluster hierarchy).

	F-measure	Time
CSA	0.638	34 m
CSA + LSA	0.702	75 m
Jaccard on *starting entities*	0.415	22 m
LSA	0.611	42 m
GED-based similarity	NA	>100 h

factor of the Personalized Page Rank (on `trec2001` we recorded analogous trends.). The length for the Contextual Graph has been considered constant and equal to 3. Like in the previous evaluation (Section 5.1), the Personalized PageRank obtains stable results between pr@30 and pr@70 and it outperforms the standard PageRank (blue dashed line) with any metrics. Moreover, the novel edge weighting function W_{TotCor} we proposed slightly outperforms the other ones, W_{Joint} and W_{Comb}.

5.3. Hierarchical document clustering

Here we evaluate CSA scalability by adapting our approach to perform hierarchical clustering on a popular benchmark dataset composed of a larger number of documents.

5.3.1. Experimental setup

We used a dataset (*re0*) of Reuters 21578,[21] a collection of 1504 manually classified documents, which is commonly used for evaluating hierarchical clustering techniques. To build the clusters hierarchy we used a hierarchical clustering algorithm, based on a similarity measure and group-average-link [36]. In this test we used only DBpedia, since was before proved that it produce more stable results.

Performance is measured in terms of goodness of fit with existing categories by using *F measure*. As defined in [37], for an entire hierarchy of clusters the F measure of any class is the maximum value it attains at any node in the tree and an overall value for the F measure is computed by taking the weighted average of all values for the F measure as given by the following: $\sum_i \frac{n_i}{n} maxF(i, j)$, where the *max* is taken over all clusters at all levels, n is the number of documents and $F(i, j)$ is the F measure for the class i and the cluster j.

5.3.2. Results and discussion

First of all, for each document d we extracted its $CG_3(d)$ and we computed $SCV(d)$ for several configurations; then, we stored bot *CGs* and *SCVs* on a file system. The whole process took just 40 min. In Table 7 a summary of the results is shown; it includes the F measures and the average of the execution time obtained running 5 time the clustering algorithm. The configuration of CSA used for obtaining these results is *GC-L*=3, W_{TotCor} and *pr*@65, which proves to be the best configuration also in this test. We produced three different baselines: Jaccard on *starting entities*, LSA [22] and GED-based (DBpedia) [10]. We considered only the GED system since it is the most similar to our approach.

As a first observation, CSA outperforms all the considered baselines in terms of F-measure and the linear combination with LSA brings a 10% improvement.

We were not able to successfully complete the test for GED due to its computational cost. Intuitively, to perform hierarchical clustering, we have to compute the inter-document similarity between all the documents of the corpus, i.e., 1501^2 measures of similarity for the *re0* dataset. While for CSA and LSA the cosine similarity is used, GED-similarity is based on a more expensive graph edit distance algorithm.

[21] Reuters collection is available at http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.

6. Conclusion and future work

In this paper, we proposed *Context Semantic Analysis* (CSA), a novel knowledge-based technique for estimating inter-document similarity. The technique is based on a Semantic Context Vector, which can be extracted from a knowledge base and stored as metadata of a document and employed to compute inter-document similarity. We showed the consistency of CSA with respect to human judges and how it outperforms standard (i.e., syntactic) inter-document similarity methods. Moreover, we obtained comparable results w.r.t. other approaches built on top of a specific knowledge base for performing semantic enrichment of the documents (i.e., ESA, WikiWalk and SSA). Our method can exploit any generic RDF *KB*. In order to evaluate CSA we employed two generic domain knowledge bases, i.e. DBpedia and Wikidata; however, CSA is applicable to a generic RDF knowledge base. To the best of our knowledge CSA is the first technique that showed its portability with two huge RDF knowledge bases. Moreover, we showed how CSA can be effectively applied in the Information Retrieval domain, even if user queries, typically composed of few words, contains a limited number of entities. We adapted CSA to be used in an existing IR framework and we showed how it can improve the performance of this framework. Finally, we experimentally demonstrate its scalability and effectiveness performing a hierarchical clustering task with a larger corpus of documents.

As future work, the proposed knowledge-based technique for inter-document similarity computation will be applied and tested in the context of *keyword searching* over relational structures [38–40]. The basic idea is to turn tuples of a relational database to documents (by considering joining and/or grouping of tuples) and then apply CSA for computing the similarity among a given document or keyword query and the documents representing the relational database. As another future work, we are planning to test the scalability of CSA also for an IR framework. A further future work, we are planning to test CSA with some domain specific knowledge bases, such as the RDF version of AGROVOC[22] and Snomed CT, respectively: an agricultural and medical knowledge base.

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, Springer, 2007.
[2] C. Bizer, T. Heath, T. Berners-Lee, Linked data - the story so far, Int. J. Semantic Web Inf. Syst. 5 (3) (2009) 1–22, http://dx.doi.org/10.4018/jswis.2009081901.
[3] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Commun. ACM 57 (10) (2014) 78–85, http://dx.doi.org/10.1145/2629489.
[4] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: A core of semantic knowledge, in: Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 697–706, http://dx.doi.org/10.1145/1242572.1242667.
[5] L. Bos, K. Donnelly, SNOMED-CT: The advanced terminology and coding system for eHealth, Stud. Health Technol. Inform. 121 (2006) 279–290.
[6] C. Caracciolo, A. Stellato, A. Morshed, G. Johannsen, S. Rajbhandari, Y. Jaques, J. Keizer, The agrovoc linked dataset, Semantic Web 4 (3) (2013) 341–348, URL http://dl.acm.org/citation.cfm?id=2786071.2786087.

[22] http://aims.fao.org/standards/agrovoc/linked-open-data.

[7] D. Beneventano, S. Bergamaschi, S. Sorrentino, M. Vincini, F. Benedetti, Semantic annotation of the cerealab database by the agrovoc linked dataset, Ecological Informatics 26 (Part 2) (2015) 119–126, Information and Decision Support Systems for Agriculture and Environment, http://dx.doi.org/10.1016/j.ecoinf.2014.07.002, URL http://www.sciencedirect.com/science/article/pii/S1574954114000843.
[8] C. Friedman, P. Kra, H. Yu, M. Krauthammer, A. Rzhetsky, Genies: a natural-language processing system for the extraction of molecular pathways from journal articles, Bioinformatics 17 (Suppl. 1) (2001) S74–S82.
[9] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, E. Motta, Semantically enhanced information retrieval: an ontology-based approach, Web Semant. Sci. Serv. Agents World Wide Web 9 (4) (2011) 434–452.
[10] M. Schuhmacher, S.P. Ponzetto, Knowledge-based graph document modeling, in: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14, ACM, New York, NY, USA, 2014, pp. 543–552, http://dx.doi.org/10.1145/2556195.2556250.
[11] P.D. Turney, P. Pantel, From frequency to meaning: Vector space models of semantics, J. Artif. Int. Res. 37 (1) (2010) 141–188, URL http://dl.acm.org/citation.cfm?id=1861751.1861756.
[12] F. Benedetti, D. Beneventano, S. Bergamaschi, Context semantic analysis: A knowledge-based technique for computing inter-document similarity, in: Similarity Search and Applications - 9th International Conference, SISAP 2016, Tokyo, Japan, October 24–26, 2016, Proceedings, 2016, pp. 164–178 http://dx.doi.org/10.1007/978-3-319-46759-7_13.
[13] F. Corcoglioniti, M. Dragoni, M. Rospocher, A.P. 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, 2016, pp. 317–333, http://dx.doi.org/10.1007/978-3-319-34129-3_20.
[14] W.H. Gomaa, A.A. Fahmy, A survey of text similarity approaches, Int. J. Comput. Appl. 68 (13) (2013) 13–18, http://dx.doi.org/10.5120/11638-7118.
[15] S.T. Dumais, Latent semantic analysis, Annu. Rev. Inf. Sci. Technol. 38 (1) (2004) 188–230.
[16] S.E. Robertson, U. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Found. Trends Inf. Retr. 3 (4) (2009) 333–389, http://dx.doi.org/10.1561/1500000019.
[17] C. Zhai, Statistical language models for information retrieval. A critical review, Found. Trends Inf. Retr. 2 (3) (2008) 137–213.
[18] E. Gabrilovich, S. Markovitch, Computing semantic relatedness using wikipedia-based explicit semantic analysis, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 1606–1611, URL http://dl.acm.org/citation.cfm?id=1625275.1625535.
[19] E. Yeh, D. Ramage, C.D. Manning, E. Agirre, A. Soroa, Wikiwalk: random walks on wikipedia for semantic relatedness, in: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, Association for Computational Linguistics, 2009, pp. 41–49.
[20] S. Hassan, R. Mihalcea, Semantic relatedness using salient semantic analysis, in: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI'11, AAAI Press, 2011, pp. 884–889, URL http://dl.acm.org/citation.cfm?id=2900423.2900564.
[21] J. Waitelonis, C. Exeler, H. Sack, Enabled generalized vector space model to improve document retrieval, in: Proceedings of the Third NLP & DBpedia Workshop (NLP & DBpedia 2015) co-located with the 14th International Semantic Web Conference 2015, ISWC 2015, Bethlehem, Pennsylvania, USA, October 11, 2015, 2015, pp. 33–44, URL http://ceur-ws.org/Vol-1581/paper4.pdf.
[22] P. Nakov, A. Popova, P. Mateev, Weight functions impact on LSA performance, in: EuroConference RANLP, 2001, pp. 187–193.
[23] R. Cyganiak, D. Wood, M. Lanthaler, RDF 1.1 concepts and abstract syntax, W3C Recommendation 25 (2014) 1–8.
[24] L. Page, S. Brin, R. Motwani, T. Winograd, The Pagerank Citation Ranking: Bringing Order to the Web, Technical Report 1999-66, Stanford InfoLab, 1999, URL http://ilpubs.stanford.edu:8090/422/.
[25] T.H. Haveliwala, Topic-sensitive pagerank, in: Proceedings of the 11th International Conference on World Wide Web, WWW '02, ACM, New York, NY, USA, 2002, pp. 517–526, http://dx.doi.org/10.1145/511446.511513.
[26] W. Xing, A. Ghorbani, Weighted pagerank algorithm, in: Proceedings. Second Annual Conference on Communication Networks and Services Research, IEEE, 2004, pp. 305–314, http://dx.doi.org/10.1109/DNSR.2004.1344743.
[27] F. Benedetti, S. Bergamaschi, L. Po, Online index extraction from linked open data sources, in: Proc.of the LD4IE Workshop 2014 co-located with the ISWC 2014.
[28] F. Benedetti, S. Bergamaschi, L. Po, Exposing the underlying schema of lod sources, in: Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2015 IEEE/WIC/ACM International Joint Conferences, 2016.
[29] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Linguist. Investig. 30 (1) (2007) 3–26, URL http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002.
[30] P.N. Mendes, M. Jakob, A. García-Silva, C. Bizer, Dbpedia spotlight: Shedding light on the web of documents, in: Proceedings of the 7th International Conference on Semantic Systems, I-Semantics '11, ACM, New York, NY, USA, 2011, pp. 1–8, http://dx.doi.org/10.1145/2063518.2063519.
[31] K. Anyanwu, A. Maduko, A. Sheth, Semrank: Ranking complex relationship search results on the semantic web, in: Proceedings of the 14th International Conference on World Wide Web, WWW '05, ACM, New York, NY, USA, 2005, pp. 117–127, http://dx.doi.org/10.1145/1060745.1060766.
[32] T. Van de Cruys, Two multivariate generalizations of pointwise mutual information, in: Proceedings of the Workshop on Distributional Semantics and Compositionality, DiSCo '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 16–20.
[33] M.D. Lee, M. Welsh, An empirical evaluation of models of text document similarity, in: Proceedings of the XXVII Annual Conference of the Cognitive Science Society, CogSci, Erlbaum, 2005, pp. 1254–1259. URL http://www.psych.unito.it/csc/cogsci05/frame/poster/2/f115-lee.pdf.
[34] D. Bär, T. Zesch, I. Gurevych, A reflective view on text similarity, in: Recent Advances in Natural Language Processing, RANLP 2011, 12–14 September, 2011, Hissar, Bulgaria, 2011, pp. 515–520, URL http://www.aclweb.org/anthology/R11-1071.
[35] E.M. Voorhees, Overview of TREC 2001, in: Proceedings of The Tenth Text REtrieval Conference, TREC 2001, Gaithersburg, Maryland, USA, November 13–16, 2001, 2001, URL http://trec.nist.gov/pubs/trec10/papers/overview_10.pdf.
[36] C.D. Manning, P. Raghavan, H. Schütze, et al., Introduction to information retrieval, vol. 1, Cambridge university press, Cambridge, 2008.
[37] Y. Zhao, G. Karypis, Evaluation of hierarchical clustering algorithms for document datasets, in: Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02, ACM, New York, NY, USA, 2002, pp. 515–524, http://dx.doi.org/10.1145/584792.584877.
[38] S. Bergamaschi, E. Domnori, F. Guerra, R. Trillo Lado, Y. Velegrakis, Keyword search over relational databases: A metadata approach, in: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, SIGMOD '11, ACM, New York, NY, USA, 2011, pp. 565–576, http://dx.doi.org/10.1145/1989323.1989383.
[39] S. Bergamaschi, F. Guerra, M. Interlandi, R.T. Lado, Y. Velegrakis, QUEST: A keyword search system for relational data based on semantic and machine learning techniques, PVLDB 6 (12) (2013) 1222–1225, URL http://www.vldb.org/pvldb/vol6/p1222-guerra.pdf.
[40] S. Bergamaschi, F. Guerra, M. Interlandi, R.T. Lado, Y. Velegrakis, Combining user and database perspective for solving keyword queries over relational databases, Inf. Syst. 55 (2016) 1–19, http://dx.doi.org/10.1016/j.is.2015.07.005.

Further Reading

[1] G. Amati, C.J. van Rijsbergen, Probabilistic models of information retrieval based on measuring the divergence from randomness, ACM Trans. Inf. Syst. 20 (4) (2002) 357–389.
[2] İ Kocabaş, B.T. Dinçer, B. Karaoğlan, A nonparametric term weighting method for information retrieval based on measuring the divergence from independence, Inf. Retr. 17 (2) (2014) 153–176.
[3] S. Clinchant, E. Gaussier, Information-based models for ad hoc IR, in: F. Crestani, S. Marchand-Maillet, E.N. Efthimiadis, J. Savoy (Eds.), Proc. 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, ACM Press, New York, USA, 2010, pp. 234–241.
[4] T. Roelleke, Information Retrieval Models. Foundations and Relationships, Morgan & Claypool Publishers, USA, 2013.
[5] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, USA, 1983.