Context Semantic Analysis: a knowledge-based technique for computing inter-document similarity

Fabio Benedetti, Domenico Beneventano, Sonia Bergamaschi

Dipartimento di Ingegneria Enzo Ferrari - Università di Modena e Reggio Emilia - Italy
firstname.lastname@unimore.it

Abstract. We propose a novel knowledge-based technique for inter-document similarity, 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. Our technique relies on a generic RDF knowledge base (e.g. DBpedia and Wikidata) to extract from it a vector able to represent the context of a document. We show how such a *Semantic Context Vector* can be effectively exploited to compute inter-document similarity. Experimental results show that our general technique outperforms baselines built on top of traditional methods, and achieves a performance similar to the ones of specialized methods.

Keywords: knowledge graph, knowledge base, inter-document similarity, similarity measures

1 Introduction

Recent years have seen growing number of knowledge bases that have been used in several domains and applications. Besides DBpedia [2], which is the heart of the Linked Open Data (LOD) cloud [5], other important examples includes: Wikidata [25], a collaborative knowledge base; YAGO [22], a huge semantic knowledge base, derived from Wikipedia, WordNet and GeoNames; Snomed CT [6], the best known ontology in the medical domain and AGROVOC [7], a multilingual agricultural thesaurus we used recently for annotating agricultural resources [4].

Recent research trends indicate that semantic information and knowledge-based approaches can be used effectively for for improving existing techniques, as Natural Language Processing (NLP) and Information Retrieval (IR); on the other hand, much still remains to be done in order to effectively exploit these rich models in these fields [21]. For instance, in the context of inter-document similarity, which plays an important role in many NLP and IR tasks, the classic techniques rely solely on syntactic information and are usually based on Vector Space Models [23], where the documents, composed by words, are represented in a vector space having words as dimensions. Such techniques fail in detecting relationships among concepts like in these two 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 which can be found by exploiting the knowledge and network structure encoded within knowledge bases such as DBpedia, even if they are not contained explicitly in the text.

In this paper, we present *Context Semantic Analysis* (CSA), a novel technique for estimating inter-document similarity, leveraging the information contained in a knowledge base. One of the main novelty of CSA w.r.t. other knowledge-based techniques for document similarity is its applicability to generic RDF knowledge bases, so that all datasets belonging to the LOD cloud [5] (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 which contains the contextual information of the document. 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*, which represents the *context* of the document. Finally, we estimate the similarity of two documents by comparing their Semantic Context Vectors with standard methods, such as the *cosine similarity*. By evaluating our method on a standard benchmark for document similarity (which consider correlations with human judges), we show how it outperforms almost all other methods and how it is portable to generic knowledge bases. Moreover we analyze and show its scalability in a clustering task with a larger corpus of documents.

The paper is structured as follows. Section 2 contains 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 contains its evaluation. Finally, the last Section contains some conclusions.

2 Related Work

Text similarity has been one the main research topic of the last few years due to wide range of its applications in tasks such as information retrieval, text classification, document clustering, topic detection, etc [11]. 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.

The standard document representation technique is the *Vector Space Model* [23]. 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) [9], 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.

Recently, a lot of effort has been employed in designing new techniques for text similarity which use information contained in knowledge bases. A first example of this *knowledge enriched* approaches is Explicit Semantic Analysis (ESA) [10], which indexes documents with Wikipedia concepts and it uses Wikipedia hyperlink structure information for mapping any text as a weighted vector of Wikipedia-based concepts. Another documents similarity technique that leverage the information contained in Wikipedia is WikiWalk [27], 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 [12]. 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 group 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 generic approach that can use use any knowledge bases expressed according to the Semantic Web standard, i.e described in RDF, so that all datasets belonging to the Linked Open Data cloud [5] (more than one thousand) can be used as source of knowledge. To the best of our knowledge, the only approach portable to generic knowledge bases is the one proposed in [21], where the authors represent documents belonging to a corpus as graphs extracted form a generic 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.

3 Preliminaries

3.1 Inter-Document 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 by 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 its 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* $\boldsymbol{d_j} = [t_{1j}, ..., t_{mj}]$. Different strategies of weighting exist (see, for example, [19]); where the weight t_{ij} is equal to the number of time the term i appears in the document j. the most famous weighting strategy is *td-idf* (*Term Frequency - Inverse Document Frequency*) [19].

The most common way of estimating the similarity of two documents is the *cosine similarity*, i.e., the cosine or angular distance between context vectors representing the two documents, because it has been shown to be effective in practice for many information retrieval applications [9].

3.2 Knowledge base

We focus on RDF knowledge bases[1]; an RDF KB can be considered a set of facts (statements), where each fact is a triple of the form <*subject*,*predicate*,*object*>. 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 them. According to [8], vertices are divided in 3 disjoint sets, URIs U, blank nodes B and literals L; literals cannot be the subjects of RDF triples.

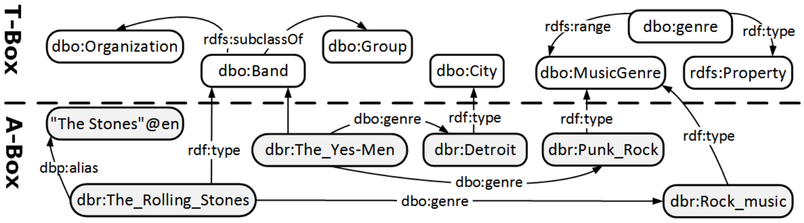

Fig. 1: Example of an RDF KB, with the *A-Box* and the *T-Box*.

The triples of an RDF KB 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, Figure 1 shows an extract of DBpedia[2]. Our methods relies only on the extensional knowledge of a *KB*, i.e. only on the *A-Box*; for our experiments we choose two generic domain KBs: DBpedia [2] and Wikidata [25], due to their large coverage and variety of relationships at the extensional level.

3.3 PageRank

PageRank was first proposed to rank web pages [20], 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 the *transition probability* matrix M is a $n \times n$ matrix given by $M_{ij} = 1/d_i$ if it exists an edge from i to j and 0 otherwise, c is the *damping factor*, a scalar value between 0 and 1 and the *personalization vector* v is a $n \times 1$ uniform vector in which each element is $1/n$. Standard PageRank uses just graph topology, but many graphs, as the ones in our case, come with weights on either nodes or edges, which can be used to *personalize* the PageRank algorithm. The *Personalized PageRank* [13] 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.

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

[2] We abbreviate URI namespaces with common prefixes, see `http://prefix.cc` for details

4 Context Semantic Analysis

Given a corpus C of documents and an RDF knowledge graph KB, CSA is composed of the following three steps:

- **Contextual Graph Extraction**: a *Contextual Graph* $CG(d)$ containing the contextual information of a document d is extracted from the KB.
- **Semantic Context Vectors Generation**: the *Semantic Context Vector* $SCV(d)$ representing the context of the document d is generated analyzing its $CG(d)$.
- **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 *KB*, 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 $(U \cup B)$ and every edge E_A corresponds to an *object property*. In Figure 1 we have only 3 triples that belongs to KB_A: the ones containing the *dbo:genre* property.

The extraction of the Contextual Graph $CG(d)$ for a document d is a three-step process:

1. Starting Entities Identification: the entities of KG_A which are explicitly mentioned in the document d are identified: such set of entities is called *starting entities* of d, denoted by $SE(d)$. The problem to find the set $SE(d)$ is an instance of the well-known *Named Entity Recognition* problem [18]; it is out of scope of this work, we tested some of the already implemented techniques and on the basis of the obtained results, we empirically chosen DBpedia Spotlight [17] 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 KG_A composed by all the triples that connect 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:

$$CG_l(d) = \{< s, p, o > \mid < s, p, o > \in KG_A \wedge < s, p, o > \in Path(s_1, s_2) \wedge \\ length(Path(s_1, s_2)) \leq l \wedge s_1, s_2 \in SE(d) \wedge s_1 \neq s_2\}$$

where $Path(s_1, s_2)$ is a path on KG_A from s_1 and s_2.
For example, let us consider the two sentences used in the introduction:
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".*

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

It is easy to find as starting entities in DBpedia: $SE(d_1)\{$**The Rolling Stones**, **Roger Daltrey**,**Trafalgar Square**$\}$ and $SE(d_2)\{$**Mick Jagger**, **The Who**,**London** $\}$. For 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 Figure 2.

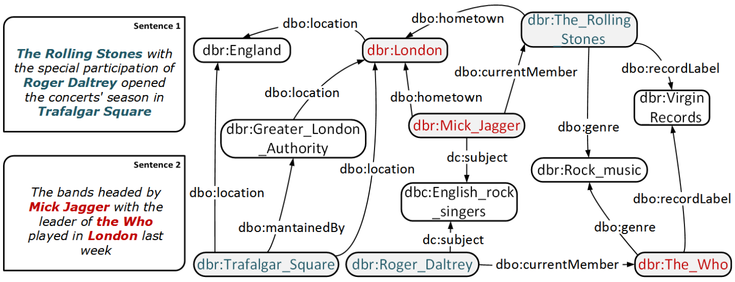

Fig. 2: Portion of DBpedia containing the most significant shared contextual information between the two sentences on the left

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 [21,1]. 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; thus, $IC(p_i)$ 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, $IC(obj_i|p_i)$ is computed in a similar way, where $P(obj_i|p_i)$ is the conditional probability that a node obj_i appears as object of the property p_i. This metric aims to provide an high weight to uncommon properties that points to uncommon object; the drawback is that it penalize 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 [21] propose to mitigate this problem by computing the *Joint Information Content* $w_{jointIC} = IC(obj_i|p_i) + IC(p_i)$, and the *Combined Information Content* $w_{combIC} = IC(obj_i) + IC(p_i)$, making an independence assumption between property and object.

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. For example, in Figure 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 $< s_i, p_i, o_i >$, 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* [24], 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].

Definitely, for contextual graphs we have three edge weights: *Total Correlation* (W_{TotCor}), Joint Information Content (W_{Joint}), and Combined Information Content (W_{Comb}).

4.2 Semantic Context Vectors

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 $C = \{d_1, ..., d_n\}$ and an RDF KB, for each document $d \in C$ we build its contextual graph $CG_l(d)$; then we consider the set $E = \{e_1, ..., e_m\}$ of entities occurring in all the contextual graphs. Similar to the *term-document matrix* (see Section 3.1) we consider an *entity-document matrix* M, 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 $SCV(d_j)$:

$$SCV(d_j) = (s(e_1, d_j), ..., s(e_m, d_j))$$

The weighting function $s(e_i, d_j)$ has to take into account for the importance of the entity e_i within $CG(d_j)$, by also considering the edge weights computed in the previous section. For this reason, we used the PageRank methods resumed in Section 3.3.
The *Semantic Context Vector* $SCV(d)$ of a document d is thus defined by 4 parameters:

1. *KB* : the RDF Knowledge Base; for example *KB=Dbpedia* and *KB=Wikidata*;
2. *CG-L* : the length for the Contextual Graph $CG_l(d)$; we used $l = 2$ and $l = 3$
3. *WeightMethod*: the edge weighting method for $CG_l(d)$: W_{Comb}, W_{Joint} and W_{TotCor}. Edge weights are used to set up the transition probability matrix M as a $k \times k$ matrix, where k is the number of nodes of $CG(d_j)$: $M_{pq} = \frac{w(p,q)}{\sum_{z=1}^{k} w(p,z)}$, where $w(p, q)$ returns the weight if an edge from p to q exists, otherwise it return 0.
   We denote with $W_{noweight}$ the case when edge weights are not used and the Standard PageRank algorithm is considered, where 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).
4. *PageRankConfiguration*: the used *damping factor* and personalization vector.
   As *damping factor* we consider a range of values from 0.10 to 0.95 with a step of 0.05. As *personalization vector* we consider the following two cases:

[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.

(a) *Standard PageRank*: in this case (denoted by *r*) there is no personalization vector, i.e., an uniform vector is considered;

(b) *Personalized PageRank*: in this case (denoted by *pr*) the personalization vector $\boldsymbol{v} = (v_1, ..., v_k)$ is setup to give an equal probability to starting entities: $v_i = 1/|SE(d)|$ if $e_i \in SE$ and 0 otherwise.

With r@50 and pr@50 we denote Standard and Personalized PageRank, respectively, with a damping factor equal to .5; the same for other damping factor values.

As an example, for the documents d_1 and d_2 of Figure 2, part of their *SCVs* are shown in Table 1; the KB is DBpedia and *CG-L* is egual 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.

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 1: Semantic Context Vectors of the two documents in Figure 2

4.3 CSA Similarity

The *CSA Similarity* between two documents d_1 and d_2 is computed as the *cosine similarity* between the Semantic Contextual Vectors $SCV(d_1)$ and $SCV(d_2)$; it is clear from the Semantic Context Vectors shown in Table 1 how the cosine similarity can detect some similarity between these two documents. In the next Evaluation Section we will analyze how the SCV's parameters affect the CSA similarity.

Linear combination of CSA with text similarity measures The CSA similarity, sim_{CSA}, is only based on information extracted from a knowledge base; we used a linear combination of the CSA similarity with other similarity measures sim_{text} (such as LSA [15] and ESA [3]) to include in the final similarity measure also textual information:

$$sim_o = \alpha * sim_{CSA} + (1 - \alpha) * sim_{text}$$

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

5 Evaluation

We evaluate the CSA performance in two different context: by considering its correlation with human judges, and, by analyzing its scalability in a clustering task.

5.1 Evaluation - Correlation with human judges

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[5] [15], which contains 50 documents, selected from the Australian Broadcasting Corporation's news mail service, evaluated by 83 students of the University of Adelaide. The performance score is given by the *Pearson product-moment correlation* coefficient r [14] between the computed similarities and the ones assigned by human judges; the Pearson coefficient r measures the linear correlation between two variables.

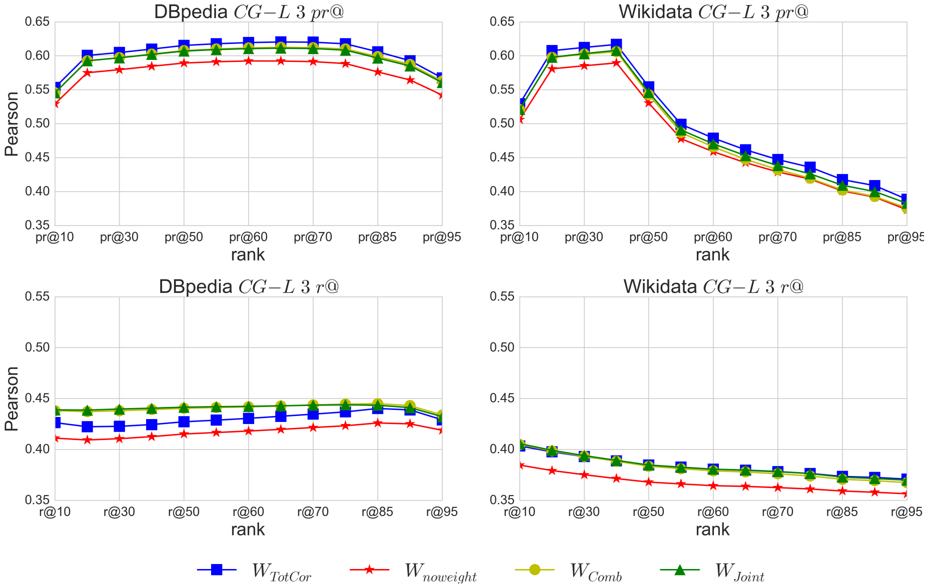

Fig. 3: Pearson correlation with human judgments (LP50 Dataset) of CSA, with different configurations.

Results and discussion A summary of the results is shown in Figure 3, 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 result 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}

[5] `https://webfiles.uci.edu/mdlee/LeePincombeWelsh.zip`

Table 2: 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

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 2 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[6].

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[7]; this improvement is particularly significant because it is only due to information extracted from the KBs by CSA[8]. 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 3 shows the performance of the linear combination of CSA with the standard text similarity measures un-backgrounded LSA [15] [9] and ESA reimplemented [3]. The best performance is obtained with $\alpha = 0.5$, and we can observe that the best configurations obtained in Table 2 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.

[6] Implemented as in [15] (only removing the stopwords)

[7] If not explicitly stated all the difference in performance are statistically significant at *p-value* < 0.05 using Fisher's Z-value transformation

[8] the sets of starting entities are obtained by using NER APIs

[9] with *td-idf* as weighting function

Table 3: 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	$pr@70$	0.67	$pr@70$	0.67	$pr@70$	0.65
	CSA + ESA	$pr@80$	0.71	$pr@65$	**0.72**	$pr@65$	0.68
Wikidata	CSA + LSA	$pr@40$	0.67	$pr@40$	0.68	$pr@40$	0.65
	CSA + ESA	$pr@40$	**0.72**	$pr@40$	**0.72**	$pr@40$	0.67

Finally, in Table 4, CSA is compared with other literature techniques. The original performance of ESA reported in [10] on the LP50 dataset has been criticized in [3] 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 [3] and [12] by re-implementing ESA without adapting the cut-off value.

Table 4: System comparison on the LP50 dataset

	Pearson coefficient r
CSA	0.62
CSA + LSA	0.65
CSA + ESA	0.72
Bag-of-Words [15]	0.41
Un-Backgrounded LSA[15]	0.52
Backgrounded LSA [15]	0.59
ESA original [10]	0.72
ESA reimplemented [3]	0.59
GED-based (Dbpedia) [21]	0.63
SSA [12]	0.68
WikiWalk + ESA [27]	0.77

The main result of such comparison is that our CSA method is able to produce results comparable with well known techniques, like LSA and ESA, and it is able to achieve improvements when it is used in conjunction with them (for example, CSA + ESA obtains a correlation $r = 0.72$, so it attains a 16% improvement). The Graph Edit Distance (GED) based approach of [21], 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.

5.2 Scalability evaluation - hierarchical document clustering

The goal of this evaluation is to estimate both the effectiveness and efficiency of CSA in a benchmark composed of a larger number of documents.

Experimental setup We used a dataset (*re0*) of Reuters 21578[10], 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 [16]. 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 [28], 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.

Table 5: 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	>100h

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[11]. The whole process took 32 hours, but we did not focus on improving the performance of this step, indeed, we can think of it as a preprocessing step. In Table 5 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 [19] and GED-based (DBpedia) [21]. 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.

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

[11] We executed this experiment in a Ubuntu machine with 16 cores (Intel Xeon E312xx) and 98 Gb of RAM

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.

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 used, when needed, for computing the Context Similarity with other documents. We showed the consistency of CSA respect to human judges and how it outperforms standard similarity methods. Moreover, we obtained comparable results w.r.t. other knowledge enriched approaches built on top of a specific KB (ESA, WikiWalk and SSA) with the advantage that our method is portable to any generic RDF *KB* (to the best of our knowledge CSA is the first system that shown its portability with two huge RDF KBs). Finally, we demonstrate its scalability and effectiveness performing hierarchical clustering with a larger corpus of documents.

To analyze the properties of CSA and to evaluate its performance we used two generic domain KBs, i.e. DBpedia and Wikidata; however, CSA is applicable to a generic RDF knowledge base. As a first future work, we are planning to test CSA with some domain specific KBs, such as the RDF version of AGROVOC[12] and Snomed CT. Then, we will analyze the time complexity needed to compute the Context Vector for any given document in order to judge the capability of CSA of dealing with web scale datasets in real/interactive time.

References

1. K. Anyanwu, A. Maduko, and A. Sheth. SemRank: ranking complex relationship search results on the semantic web. In *Proceedings of the 14th international conference on World Wide Web*, pages 117–127. ACM, 2005.
2. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *DBpedia: A nucleus for a web of open data*. Springer, 2007.
3. D. Bär, T. Zesch, and I. Gurevych. A reflective view on text similarity. In *RANLP*, pages 515–520, 2011.
4. D. Beneventano, S. Bergamaschi, S. Sorrentino, M. Vincini, and F. Benedetti. Semantic annotation of the cerealab database by the agrovoc linked dataset. *Ecological Informatics*, 26:119–126, 2015.
5. C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.
6. L. Bos and K. Donnelly. SNOMED-CT: The advanced terminology and coding system for eHealth. *Stud Health Technol Inform*, 121:279–290, 2006.
7. C. Caracciolo, A. Stellato, A. Morshed, G. Johannsen, S. Rajbhandari, Y. Jaques, and J. Keizer. The AGROVOC linked dataset. *Semantic Web*, 4(3):341–348, 2013.

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

8. R. Cyganiak, D. Wood, and M. Lanthaler. RDF 1.1 concepts and abstract syntax. *W3C Recommendation*, 25:1–8, 2014.
9. S. T. Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.
10. E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.
11. W. H. Gomaa and A. A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13), 2013.
12. S. Hassan and R. Mihalcea. Semantic relatedness using salient semantic analysis. In *AAAI*, 2011.
13. T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002.
14. I. Lawrence and K. Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pages 255–268, 1989.
15. M. Lee, B. Pincombe, and M. Welsh. An empirical evaluation of models of text document similarity. *Cognitive Science*, 2005.
16. C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
17. P. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia Spotlight: Shedding light on the web of documents. I-Semantics, 2011.
18. D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.
19. P. Nakov, A. Popova, and P. Mateev. Weight functions impact on LSA performance. *Euro-Conference RANLP*, pages 187–193, 2001.
20. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.
21. M. Schuhmacher and S. P. Ponzetto. Knowledge-based graph document modeling. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 543–552. ACM, 2014.
22. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.
23. P. D. Turney, P. Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.
24. T. Van de Cruys. Two multivariate generalizations of pointwise mutual information. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 16–20. Association for Computational Linguistics, 2011.
25. D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.
26. W. Xing and A. Ghorbani. Weighted pagerank algorithm. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 305–314. IEEE, 2004.
27. E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. Wikiwalk: random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics, 2009.
28. Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM, 2002.