Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

REWOrD: Semantic Relatedness in the Web of Data

Giuseppe Pirró
KRDB, Free University of Bolzano-Bozen, Italy
giuseppe.pirro@unibz.it

Abstract

This paper presents REWOrD, an approach to compute semantic relatedness between entities in the Web of Data representing real word concepts. REWOrD exploits the graph nature of RDF data and the SPARQL query language to access this data. Through simple queries, REWOrD constructs weighted vectors keeping the *informativeness* of RDF predicates used to make statements about the entities being compared. The most informative path is also considered to further refine informativeness. Relatedness is then computed by the cosine of the weighted vectors. Differently from previous approaches based on Wikipedia, REWOrD does not require any preprocessing or custom data transformation. Indeed, it can leverage whatever RDF knowledge base as a source of background knowledge. We evaluated REWOrD in different settings by using a new dataset of real word entities and investigate its flexibility. As compared to related work on classical datasets, REWOrD obtains comparable results while, on one side, it avoids the burden of preprocessing and data transformation and, on the other side, it provides more flexibility and applicability in a broad range of domains.

1 Introduction

The ability to estimate relatedness lies at the core of cognition and is an important pillar in processes such as memory, categorization, decision making, problem solving, and reasoning (Schaeffer and Wallace 1969). In this paper we focus on the following problem: given two words, compute to what extent they are related form a semantic point of view. Relatedness is exploited in many domains ranging from NLP, where it is used to compare texts (Mihalcea, Corley, and Strapparava 2006) or perform word sense disambiguation (Patwardhan, Banerjee, and Pedersen 2003)), to biomedical informatics where relatedness is used to compare medical terms (Pedersen et al. 2007).

To emulate as much as possible the human behavior, when computing relatedness it is necessary to lift words to the level of concepts i.e., complex representations derived by some kind of inference triggered by the words themselves. Working with concepts enables to exploit relations that these concepts bear with others and, more important, their semantics. As an example, while the words *Turing* and *Computer Science* are unrelated if treated as strings, their lifting to the level of concepts enables to discover perspectives (i.e., relations) relating the original words. What is needed is some form of background knowledge.

Computational approaches to relatedness have considered different forms of background knowledge, e.g., handcrafted lexical ontologies like WordNet (Resnik 1995; Budanitsky A 2001), search engine indexes (Bollegala, Matsuo, and Ishizuka 2007; Turney 2001) or large corpora (Landauer and Dumais 1997). Although approaches à la WordNet feature accurate knowledge, they are limited in terminology coverage and require a significant creation and maintenance effort. On the other hand, corpus-based techniques provide a broader coverage but information is not structured. A good trade-off is Wikipedia, where knowledge in different domains is manually curated by thousands of contributors. Recently, many approaches started to consider knowledge in Wikipedia for relatedness estimation. For instance, ESA (Gabrilovich and Markovitch 2007), constructs an index of the whole text of Wikipedia and vectors containing the relevance of articles, representing concepts, w.r.t. each word appearing in Wikipedia. WikiWalk! (Yeh et al. 2009) leverages PageRank-like scores assigned to articles by preprocessing the whole structure of Wikipedia. Interestingly, these techniques obtain more accurate results than those based on WordNet, search engines or Latent Semantic Analysis as reported in (Gabrilovich and Markovitch 2007).

Current approaches to relatedness are too knowledge-source-specific and require a significant preprocessing and data transformation effort, which may hinder their scalability when adopting a different source of knowledge. Besides, although some approaches based on Wikipedia look at the link structure, they do not consider connectivity between the entities being compared. Finally, none of these approaches is grounded on Semantic Web technologies and languages although a large amount of background knowledge is now encoded in RDF and published in the Web of Data (Heath and Bizer 2011). A notable example is DBPedia (Auer et al. 2008), the Web of Data counterpart of Wikipedia, which encodes facts about 8M of entities and their semantic relations in 1 billion of *RDF triples*. It is interesting to observe that the huge amount of knowledge in DBPedia and other RDF knowledge bases can be accessed through powerful query languages like SPARQL without any preprocessing.

Contributions. The contributions of this paper are as follows. We propose REWOrD, a method for computing relatedness between Web of Data entities representing real world concepts. REWOrD exploits Web of Data knowledge sources such as DBPedia and relies on the SPARQL query language to access data on-the-fly with the aim to construct weighted vectors containing the *informativeness* of RDF predicates used to make statements about the entities being compared. REWOrD also looks at paths between these entities to further refine their relatedness. We collected a new dataset of 26 pairs of entities for which relatedness judgements have been provided by 20 judges. The motivation for this new dataset is that existing datasets contain couples of generic entities such as *Car-Automobile* and miss couples of more specific entities such as *Android-Linux*. We evaluated REWOrD on different scenarios by using DBPedia. To investigate its flexibility, we considered another set of entities defined both in a general source of background knowledge (i.e., DBPedia) and in more specific one (i.e., LinkedMDB). Finally, by using existing datasets we compared REWOrD with related work. REWOrD obtains results comparable with the state of the art even if: *i)* it avoids the burden of preprocessing and data transformation and; *ii)* it is more flexible and widely applicable as it is grounded on Semantic Web technologies and languages.

2 Preliminaries

This section provides some background on RDF, presents the underlying data model used by REWOrD and briefly introduces the features of SPARQL exploited by REWOrD.

RDF and relatedness. Given a set of URIs $\mathcal{U}$ and a set of literals $\mathcal{L}$, an *RDF triple* is defined as: $\langle s,p,o\rangle$, where $s \in (\mathcal{U})$ is the *subject*, $p \in \mathcal{U}$ is the *predicate* (or *property*), and $o \in (\mathcal{U} \cup \mathcal{L})$ is the *object*[1]. A triple states that the property p holds between the subject s and the object o thus identifying a precise perspective on which the subject and the object are *related*. Hence, a triple can be seen as the finest-grained level of relatedness between the subject and the object. For instance, the triple $\langle$dbp:EnricoFermi, dbp:birth-place, dbp:Italy$\rangle$ relates *Enrico Fermi* to *Italy* from the perspective that he was born in *Italy*. Different predicates enable a multidimensional *relatedness space* where each dimension (i.e., RDF predicate) covers a particular relatedness perspective.

A triple can be graphically represented by two nodes (the subject and the object) and a directed edge (representing the predicate) from the subject to the object node. A collection of RDF triples forms an *RDF graph*.

Data model. We consider RDF knowledge bases as source of background knowledge for estimating relatedness. An RDF knowledge base can be seen as a directed multi graph $\mathcal{G}_w = \langle \mathcal{V}, \mathcal{E}, \mathcal{D}, \mathcal{F}\rangle$ where $\mathcal{V}$ is a finite set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ if a finite set of edges. We denote with $e(v_i, v_j)$ an edge from node v_i to node v_j. $\mathcal{D}$ is a description function $\mathcal{D} : v \mapsto P(\mathcal{T})$ mapping a node $v \in \mathcal{V}$ to a subset of the power set of $\mathcal{T}$, which contains all possible triples in $\mathcal{G}_w$. The function $\mathcal{F}$: e $\mapsto \mathcal{S}$ associates to each edge a label from a finite set $\mathcal{S}$. Note that $\mathcal{V}$ may contain both URIs and literals. The function $\mathcal{D}$ simulates a SPARQL query, which enables to obtain a set of triples about a node v representing a URI. The function $\mathcal{F}$ specifies the edge type, that is, the RDF predicate belonging to the relatedness space $\mathcal{S}$ containing all the predicates. A path between the nodes $v_0 \in \mathcal{U}$ and $v_n \in \mathcal{U}$ in $\mathcal{G}_w$ is defined as $\mathrm{p}(v_0, v_n) = v_0 \overset{e_1}{-} v_1 \overset{e_2}{-} ... v_{n-1} \overset{e_m}{-} v_n$ with $v_i \in \mathcal{V}\ \forall i \in [0,n]$, $e_i = (v_{j-1}, v_j) \in \mathcal{E}\ \forall j \in [1,m]$ and $- \in \{\leftarrow, \rightarrow\}$. The symbol $-$ models the fact that in a path we may have edges pointing in different directions. As an example, in the graph depicted in Fig. 1 we have p(dbp:Enrico_Fermi, dbp:Hideki_Yukawa) = dbp:Enrico_Fermi $\overset{e_1}{\leftarrow}$ dbp:Hideki_Yukawa where e_1 = dbp-owl:influencedBy.

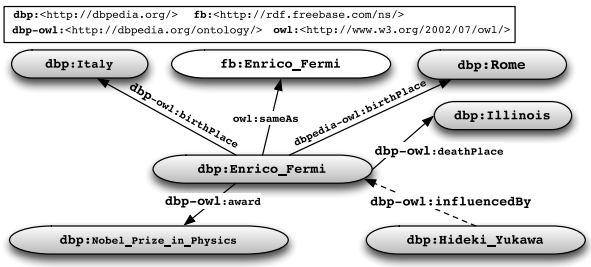


Figure 1: An excerpt of the RDF graph associated to dbp:Enrico_Fermi in DBPedia.

SPARQL. SPARQL is the W3C language for querying RDF data. It features a set of constructs very similar to those provided by SQL for relational databases. The more recent SPARQL 1.1. provides aggregate operators and path expressions among the other things. In this paper we are interested in the most basic form of SPARQL queries that is Basic Graph Patterns (BGP). Besides, we will make usage of aggregate operators such as COUNT. Given a SPARQL query q, a solution to q is defined in terms of the matching of a BGP b, which can be seen as a subgraph, in the queried RDF graph. According to the SPARQL specification[2] query variables are *bound* to RDF terms (i.e, URIs and literals) via a solution mappings μ that can be seen as a set of variable-term-pairs with each variable not appearing in more than a pair. In more detail, the application of a solution mapping μ to a BGP b (i.e., $\mu[b]$) means that each variable in the BGP b, which is bound, is replaced by the RDF term in μ. Besides, variables that are not bound must not be replaced. As an example, the query SELECT ?*pl* WHERE {dbp:Enrico_Fermi dbp-owl:BirthPlace ?*pl*} executed over the graph in Fig. 1 will *bind* the variable ?*pl* to dbp:Rome. On the other hand, the query SELECT COUNT(?*p*) as ?*count* WHERE {dbp:Enrico_Fermi ?*p* ?*o*} counts the number of triples in which dbp:Enrico_Fermi is the subject.

[1] We do not consider the case in which s and o are *blank nodes* as their usage is discouraged (Heath and Bizer 2011).

[2] http://www.w3.org/TR/sparql11-query/

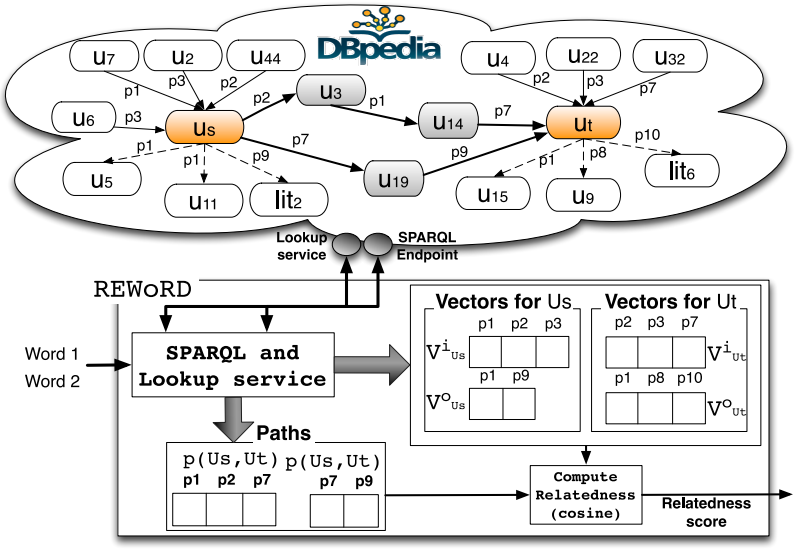


Figure 2: An overview of REWOrD.

3 REWOrD: An overview

This section provides an overview of REWOrD. A more detailed discussion will the subject of the subsequent sections. As considered before, an RDF predicate represents the finest-grained level for expressing relatedness between the subject and the object of a triple. As it is possible to have several kinds of predicates, one can imagine a relatedness space $\mathcal{S}$ with each dimension (i.e., predicate) covering a particular relatedness perspective from the subject toward an object. One may also think of as the inverse relation holding in the reverse direction. Having or not such relation depends on how data are defined. Identifying which dimensions are relevant for a particular URI amounts at finding the set of triples, and more specifically the predicates, in which the URI appears. The relatedness space resembles the feature space proposed by Tversky (Tversky 1977) in cognitive science. Here, likeness between objects is estimated by taking into account their "features" defined in a space of features. A feature can be thought of as a *property* of an object. As an example, having a *steering wheel* is a feature of a *car* and the underlying common-sense relation is *part-of*. In RDF this piece of information can be encoded as $\langle$ns:steering_wheel, ns:part-of, ns:car$\rangle$.

Fig. 2 depicts how REWOrD works in the Web of Data. Given two words in input, the first step is to find URIs that better correspond to the concepts triggered by these words. REWOrD relies on lookup services such as that provided by DBPedia. As an example, for the word *USA*, the look up service returns the URI http://dbpedia.org/resource/United˙States. In case of multiple URIs returned for the same word in input, REWOrD takes the one with the highest rank. However, this behavior is easily extensible to the case in which the user chooses the most appropriate URI.

The relatedness space for a URI u (i.e., $\mathcal{S}^{\mathrm{u}}$) is modelled a k-dimensional weighted vector $\mathcal{V}_{\mathrm{u}}$, where each dimension represents the *informativeness* of a predicate. To construct the relatedness space and compute the informativeness, REWOrD issues SPARQL queries to an endpoint. In the figure, we consider the case of DBPedia although any other endpoint (*Freebase*, *The New York Times*, *DBLP*, etc.) can be considered. Note that as an RDF predicate can be relevant for a URI u in two ways, that is, when u is the subject or the object of a triple, REWOrD considers two different kinds: one for the incoming predicates ($\mathcal{V}^i_{\mathrm{u}}$) and the other for the outgoing ($\mathcal{V}^0_{\mathrm{u}}$). Relatedness is computed by considering the cosine between the vectors associated to the URIs being compared augmented with information about the *most informative* path connecting them. Section 6 investigates different combinations of predicate and path informativeness.

4 Informativeness in the Relatedness Space

As it will be discussed in the Related Work section, some approaches for computing relatedness in Wikipedia (Yeh et al. 2009) exploit PageRank and similar algorithms to assign a popularity score to the nodes of the graph derived from Wikipedia articles. This requires a huge preprocessing effort since the whole graph has to be processed as the score of a node depends on the score of other nodes. Besides, even changing a single edge, which is relatively frequent in Wikipedia, can affect the PageRank score of many nodes.

REWOrD relies on a different strategy based on weighted vectors of predicates. To construct these vectors, a simple approach would be that of inserting 1 if the predicate is relevant (i.e., appears in some triple with u) for a URI u and 0 otherwise. However, this approach does not take into account the relative importance of predicates. For instance, $p3$ in Fig. 2 is used twice as an incoming predicate for u_s whereas $p1$ twice as an outgoing predicate. As it happens for words and documents in the vector space model, here it has to be considered to what extent predicates are *informative* for a particular URI. Therefore, the notion of predicate informativeness comes into play. Inspired by the TFIDF, we introduce the Predicate Frequency ($\mathcal{PF}$) Inverse Triple Frequency ($\mathcal{ITF}$) to model informativeness.

Predicate Frequency ($\mathcal{PF}$). $\mathcal{PF}$ quantifies the informativeness of a predicate p in the context of a URI $\mathtt{u}$. With context we mean the RDF triples where p and $\mathtt{u}$ appear together. Note that p may be used as an incoming or outgoing predicate w.r.t. $\mathtt{u}$; therefore we distinguish between *incoming* $\mathcal{PF}$ ($\mathcal{PF}_i^{\mathtt{u}}(p)$) and outgoing $\mathcal{PF}$ ($\mathcal{PF}_o^{\mathtt{u}}(p)$). This coefficient, which resembles the Term Frequency used in the vector space model (Salton, Wong, and Yang 1975) considers the number of times p is used with $\mathtt{u}$ as compared to the total number of predicates linking $\mathtt{u}$ with other resources or literals. In more detail we have:

$$\mathcal{PF}^{\mathtt{u}}_{x\in\{i,o\}}(p) = |T^{\mathtt{u}}(p)|/|T^{\mathtt{u}}| \,. \quad (1)$$

where $|T^{\mathtt{u}}(p)|$ denotes the number of triples of the form $\langle ?s, p, \mathtt{u}\rangle$ for $\mathcal{PF}_i^{\mathtt{u}}(p)$ (resp., $\langle \mathtt{u}, p, ?o\rangle$ for $\mathcal{PF}_o^{\mathtt{u}}(p)$) and $|T^{\mathtt{u}}|$ the total number of triples in which $\mathtt{u}$ appears.

Inverse Triple Frequency ($\mathcal{ITF}$). The inverse triple frequency $\mathcal{ITF}(p)$, considers how many times a predicate is used in some RDF triple w.r.t. the total number of triples, and is defined as:

$$\mathcal{ITF}(p) = log|T|/|T(p)| \,. \quad (2)$$

where $|T|$ is the total number of triples in the knowledge base and $|T(p)|$ the total number of triples having p as a predicate. Finally, the *incoming* (resp., *outgoing*) $\mathcal{PF}\,\mathcal{ITF}$ is defined as $\mathcal{PF}\,\mathcal{ITF}_i(p)$=$\mathcal{PF}_i \times \mathcal{ITF}$ (resp., $\mathcal{PF}\,\mathcal{ITF}_o(p)$=$\mathcal{PF}_o \times \mathcal{ITF}$).

Hence, the vectors $\mathcal{V}^i_{\mathtt{u}}$ and $\mathcal{V}^o_{\mathtt{u}}$ (see Fig. 2) will contain $\mathcal{PF}\,\mathcal{ITF}_i$ and $\mathcal{PF}\,\mathcal{ITF}_o$ values for each dimension (i.e., predicate) in the relatedness space, respectively. In order to obtain predicates relevant for a URI and predicate counts, REWOrD relies on SPARQL queries containing BGPs and the COUNT operator. Further implementation details are available at the REWOrD Web site.[3]

5 Path Informativeness

The second ingredient of REWOrD's approach to relatedness, as reported in Fig. 2, are paths between URIs. In knowledge sources such as WordNet or Mesh, which feature a tree-like structure, a natural approach to compute the relatedness between concepts is to look at paths between them (see for instance (Budanitsky A 2001)(Pirró and Euzenat 2010)(Resnik 1995)). Besides, the most specific common ancestor (*msca*) is exploited as a representative of the ratio of commonalities between the concepts (Resnik 1995).

When dealing with RDF knowledge bases such as DBPedia, since the underlying data forms a graph, the notion of *msca* does not apply. However, paths between resources are still useful to understand what these resource share. In general, an RDF triple of the form $\langle \mathtt{u}_i, p, \mathtt{u}_j\rangle$ may be thought of as a path of length 1 between the URIs $\mathtt{u}_i$ and $\mathtt{u}_j$. By generalizing this idea, a BGP of the form $\langle \mathtt{u}_j, ?p, ?d\rangle$ can be seen as a path of length 1 between $\mathtt{u}_i$ and the result of the binding of the variables $?p$ and $?d$. If this BGP is chained with another one on the variable $?d$, that is $\langle \mathtt{u}_i, ?p, \mathbf{?d}\rangle\langle \mathbf{?d}, ?p1, \mathtt{u}_j\rangle$, we are able to obtain a path of length 2 between $\mathtt{u}_i$ and $\mathtt{u}_j$ and so forth. Note that SPARQL 1.1 supports property paths, that is, a way to discover routes between nodes in an RDF graph. However, since variables can not be used as part of the path specification itself, this approach is not suitable for our purpose. Therefore, we introduce *k-BGP* reachability.

k-BGP reachability. Given an integer k and two URIs $\mathtt{u}_0$ and $\mathtt{u}_n$, k-BGP reachability, denoted by $\mathcal{R}^{BGP}(\mathtt{u}_0, \mathtt{u}_n, k)$, computes the set of paths of length at most k connecting $\mathtt{u}_0$ and $\mathtt{u}_n$. For instance, if $k = 3$ then all the paths of length 1, 2 and 3 will be considered.

Semantics. The interpretation of $\mathcal{R}^{BGP}(\mathtt{u}_0, \mathtt{u}_n, k)$ over the graph $\mathcal{G}_w$ are all the paths of the form $\mathtt{p}(\mathtt{u}_0, \mathtt{u}_n) = \mathtt{u}_0 ... \overset{p_q}{-} \mathtt{u}_n$ with $q \leq k$. We say that an edge $e(v_i, v_j) \in \mathcal{G}_w$ matches a subpath $\mathtt{u}_i \overset{p_{i+1}}{-} \mathtt{u}_j \in \mathtt{p}$ if $\mathtt{u}_i$, $\mathtt{u}_j$ and p_{i+1} are variables that have been bound in the evaluation of the $(i+1)$-BGP or constants.

Path Informativeness. For a given $q \leq k$, several paths may exist. The problem now is to identify the most informative one among them. We propose the following approach: given a path of length 1 of the form $\mathtt{p}(\mathtt{u}_s, \mathtt{u}_t) = \mathtt{u}_s \xrightarrow{p} \mathtt{u}_t$, the informativeness of the predicate p in the path $\mathtt{p}$, denoted by $I^p(\mathtt{p}(\mathtt{u}_s, \mathtt{u}_t))$ is computed as:

$$I^p(\mathtt{p}(\mathtt{u}_s, \mathtt{u}_t)) = [\mathcal{PF}\,\mathcal{ITF}^{\mathtt{u}_s}_o(p) + \mathcal{PF}\,\mathcal{ITF}^{\mathtt{u}_t}_i(p)]/2 \quad (3)$$

It is considers the predicate p as outgoing from $\mathtt{u}_s$ and incoming to $\mathtt{u}_t$. An analogue formula can be built for the case $\mathtt{p}(\mathtt{u}_s, \mathtt{u}_t) = \mathtt{u}_s \xleftarrow{p} \mathtt{u}_t$, where p is incoming in $\mathtt{u}_s$ and outgoing from $\mathtt{u}_t$. Paths of length greater than 1, can be decomposed into a set of sub-paths of length 1, for which the informativeness can be computed. The informativeness of the whole path is computed as $[I^{p1}(\mathtt{p}(\mathtt{u}_s, \mathtt{u}_k)) + ... + I^{pm}(\mathtt{p}(\mathtt{u}_r, \mathtt{u}_q)) + I^{pn}(\mathtt{p}(\mathtt{u}_q, \mathtt{u}_t))]/|\mathtt{p}|$. It is basically the sum of the informativeness of the sub-paths divided by the length of the path. Path informativeness enables to discover the most informative chain of RDF predicates, which connect $\mathtt{u}_s$ and $\mathtt{u}_t$. The informativeness of these predicates is summed to that of the predicates in the vectors $\mathcal{V}_{\mathtt{Us}}$ and $\mathcal{V}_{\mathtt{Ut}}$ (see Fig. 2). If these predicates were not present, then a new shared dimension is added in both vectors.

6 Evaluation

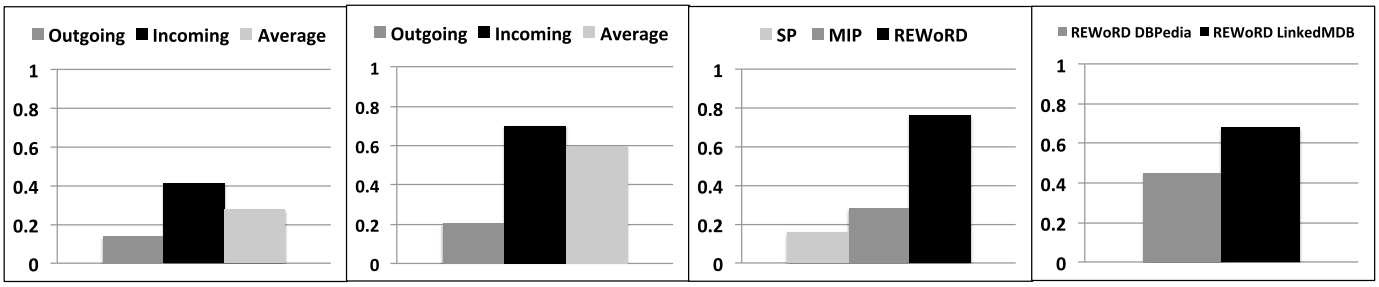
We implemented REWOrD in Java by using Jena[4]. In our experiments, the DBPedia[5] and LinkedMDB[6] SPARQL endpoints have been used. Recall that REWOrD does not need any preprocessing of data as compared to Wikipedia-based approaches such ESA, where authors reported that is was necessary to process almost 3GBs of text (Gabrilovich and Markovitch 2007). Moreover, although we used only two SPARQL endpoints in this evaluation, REWOrD can exploit any other endpoint. Finally, note that if data was locally available then SPARQL queries could be locally issued without any network communication.

[3] http://relwod.wordpress.com

[4] http://incubator.apache.org/jena

[5] http://dbpedia.org/snorql

[6] http://www.linkedmdb.org/snorql



(a) ρ with $\mathcal{PF}$ (b) ρ with $\mathcal{PF}\,\mathcal{ITF}$ (c) ρ with path info (k=3) (d) ρ on different sources (k=3)

Figure 3: Evaluation of REWOrD in different scenarios.

Datasets and evaluation methodology. To evaluate techniques for computing relatedness, a common approach is to compare the scores they provide with the scores provided by humans performing the same task. This approach provides an application-independent way for evaluating measures of relatedness. In particular, three de facto standard datasets are used: i) *R&G* (Rubenstein and Goodenough 1965) that contains 65 pairs of words; ii) *M&C* (Miller and Charles 1991) that contains 30 pairs of words; iii) WSIM-353[7] that contains 353 pairs of words. These datasets only contain couples of general entities such as *Car-Automobile* or *Planet-Sun*, while more specific couples such as *Android-Linux* are missing. Therefore, a new dataset, referred to as *G*26, has been constructed, which contains 26 pairs of entities for which relatedness judgements have been provided by 20 computer science master students on a scale from 1 to 4. Word pairs in *G*26 have been manually selected by considering different domains such as *Sport*, *Music* and *Technology*.

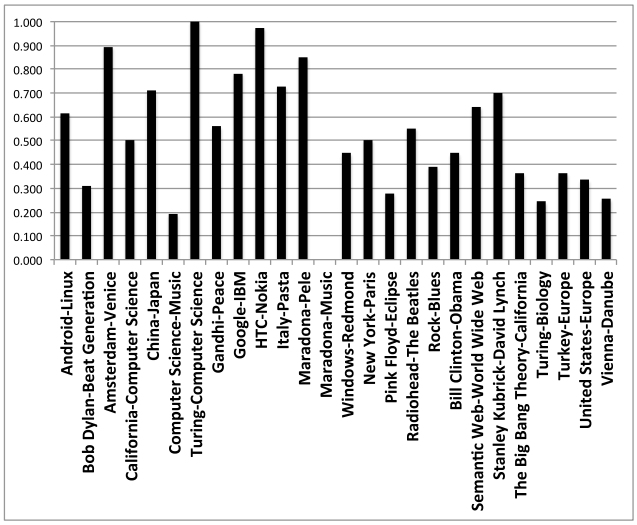


Figure 4: The *G*26 dataset.

The word pairs along with the average human judgements (normalized between 0 and 1) are reported in Fig. 4. The inter-annotator agreement reported was of 0.75. In all the experiments, the accuracy of computational methods is assessed by computing the Spearman correlation coefficient ρ between their scores and the gold standard.

Evaluation 1: evaluating REWOrD on *G*26. In the first evaluation setting only the predicate frequency $\mathcal{PF}$ has been considered in order to build the weighted vectors associated to the concepts being compared. We considered three cases: i) $\mathcal{PF}$ on incoming RDF predicates ($\mathcal{PF}^i$); ii) $\mathcal{PF}$ on outgoing predicates ($\mathcal{PF}^o$); iii) their average. Results are reported in Fig. 3 (a). As it can be observed, the $\mathcal{PF}$ alone gives poor results, especially when only outgoing RDF predicates are considered. The value of correlation is $\rho = 0.14$. $\mathcal{PF}^i$ brings better result with $\rho = 0.4$. When considering the average of the two values the correlation is of almost 0.3.

In the second evaluation setting only the $\mathcal{PF}\,\mathcal{ITF}$ has been considered. Correlation results are reported in Fig. 3 (b). Here, there is no significant improvement when considering $\mathcal{PF}\,\mathcal{ITF}^o$ instead of $\mathcal{PF}^o$ while in the case of ingoing predicates (i.e., $\mathcal{PF}\,\mathcal{ITF}^i$) there is a significant improvement, with correlation reaching the value 0.7. When considering the average of the two values the correlation is 0.6. This suggests two things. First, the informativeness of outgoing predicates is a less accurate estimator of relatedness than that of incoming predicates. Second, considering the $\mathcal{ITF}$ pays in terms of performance in both cases but with a significant improvement in the case of ingoing predicates.

We performed an additional set of tests (see Fig. 3 (c)) by considering the shortest path (*SP*) between the URIs of the concepts being compared and the most informative path (*MIP*). Besides, a combination of *MIP* with the $\mathcal{PF}\,\mathcal{ITF}^i$ was also considered. This combination is refereed to as REWOrD in Fig. 3 (c). In these experiments we looked at paths of length at most 3 (i.e., $k = 3$). As it can be noted, the *SP* strategy brings poor results. This is because in some cases there are no paths between the URIs compared. A bit of improvement can be observed when using the *MIP*, which looks at the most informative path by considering the informativeness of RDF predicates. Combining the *MIP* with the $\mathcal{PF}\,\mathcal{ITF}^i$ brings the best results with a value of correlation of about 0.76.

The last experiment was performed on a set of 10 couples of actors defined both in DBPedia and LinkedMDB. The dataset along with human ratings are available at the REWOrD

[7] http://www.cs.technion.ac.il/gabr/resources/data/wordsim353

website.[8] In this evaluation we considered the REWOrD configuration with *MIP* and $\mathcal{PFITF}^i$. As it can be observed in Fig. 3 (d), the correlation is higher when using LinkedMDB as source of knowledge. This is probably due to the fact that in LinkedMDB the set of predicates and their links to resources better capture relations among actors. Indeed, LinkedMDB is a rich source of background knowledge about movies and actors.

Evaluation 2: comparison with other approaches. In this experiment, REWOrD has been compared with related work on three commonly used datasets (see Table 1). The scores for WikiRelate! are reported in (Ponzetto and Strube 2007), for ESA in (Gabrilovich and Markovitch 2007), for WLM in (Milne and Witten 2008) and for WikiWalk in (Yeh et al. 2009). As it can be observed, REWOrD on all datasets approaches ESA while overcomes WikiRelate!.

Table 1: Correlation on existing datasets

	Spearman Correlation		
Measure	**M&C**	**R&G**	**WSIM-353**
WikiRelate!	0.45	0.52	0.49
ESA	0.73	0.82	0.75
WLM	0.70	0.64	0.69
WikiWalk	0.61	-	0.63
REWOrD	0.72	0.78	0.73

Note that ESA relies on a huge amount of text obtained by parsing the whole Wikipedia content to build an inverted index whereas REWOrD only relies on on-the-fly information obtained by querying a SPARQL endpoint. Interestingly, REWOrD performs better than other approaches based on links (i.e., WLN, WikiWalk), which require a much higher preprocessing effort. These approaches are mostly ad-hoc, in the sense that their applicability to other sources of background knowledge is not immediate. ESA was also evaluated on the Open Directory Project but even in this case, the preprocessing effort was quite high. Authors reported of having processed 3 million of URLs and 70GBs of data (Gabrilovich and Markovitch 2007).

To further compare these approaches with REWOrD it would be interesting to investigate their results on the *G*26 dataset or on other sources of background knowledge. One advantage of REWOrD over these approaches is that it is immediate to compute relatedness between entities in a new domain. What is needed is just the address of a new SPARQL endpoint. For instance, if one wanted to compute relatedness between proteins, a SPARQL endpoint such that provided by UniProt could be exploited.[9]

7 Related Work

In this paper we focused on semantic relatedness, which generalizes similarity by considering not only specialization relations between words. The application of semantic relatedness span different areas from natural language processing (Patwardhan, Banerjee, and Pedersen 2003) to distributed systems (Pirró, Ruffolo, and Talia 2008). In the Semantic Web context, some initiatives consider RDF predicates for vocabulary suggestion (Oren, Gerke, and Decker 2007) while other (Freitas et al. 2011) exploit relatedness for query answering over Linked Data. However, differently from REWOrD none of them is specifically focused on computing relatedness in the Web of Data.

Generally speaking, computational approaches to relatedness exploit different sources of background knowledge such as WordNet (e.g., (Resnik 1995; Budanitsky A 2001)), MeSH (e.g., (Rada, Mili, and Bicknell 1989; Pirró and Euzenat 2010)) or search engines (e.g., (Bollegala, Matsuo, and Ishizuka 2007; Turney 2001)). Recently, Wikipedia has been shown to be the most promising source of background knowledge for relatedness estimation (Gabrilovich and Markovitch 2007). Therefore we'll consider approaches exploiting Wikipedia as baseline for comparison.

WikiRelate! (Ponzetto and Strube 2007), given two words first retrieves the corresponding Wikipedia articles whose titles contain the words in input. Then, it estimates relatedness according to different strategies among which comparing the texts in the pages or computing the distance between the Wikipedia categories to which the pages belong.

Explicit Semantic Analysis (Gabrilovich and Markovitch 2007) compute relatedness both between words and text fragments. ESA derives an interpretation space for concepts by preprocessing the content of Wikipedia to build an inverted index that for each word, appearing in the corpus of Wikipedia articles, contains a weighted list of articles relevant to that word. Relevance is assessed by the TFIDF weighting scheme while relatedness is computed by the cosine of the vectors associated to the texts in input. WLM (Milne and Witten 2008) instead of exploiting text in Wikipedia articles, scrutinizes incoming/outgoing links to/from articles. WikiWalk (Yeh et al. 2009) extends the WLM by exploiting not only link that appear in an article (i.e., a Wikipedia page) but all links, to perform a random walk based on *Personalized PageRank*.

The most promising approach, in terms of correlation, is ESA. However, ESA requires a huge preprocessing effort to build the index, only leverages text in Wikipedia and does not consider links among articles. Therefore, it may suffer some problems when the amount of text available is not large enough to build the interpretation vectors or when changing the source of background knowledge.

REWOrD is more flexible as it only needs a SPARQL endpoint to get the necessary information. WikiRelate! looks at paths but only from the point of view of categories while REWOrD looks at the level of data. WLM looks at links among Wikipedia articles but does not consider paths connecting them. Finally, WikiWalk requires to preprocess of the whole Wikipedia graph to obtain PageRank scores. Overall, although these approaches are very promising they require a huge preprocessing effort and are not flexible since changing the source of background knowledge implies to restart a new preprocessing phase. On the other hand, REWOrD being based on Semantic Web technologies only needs to query a (local or remote) SPARQL endpoint to get the necessary background knowledge.

[8] http://relwod.wordpress.com

[9] http://uniprot.bio2rdf.org/sparql

8 Concluding Remarks and Future Work

This paper presented REWOrD an approach to compute relatedness exploiting SPARQL to construct vectors containing the informativeness of RDF predicates used to make statements about the concepts being compared. Informativeness is computed by the Predicate Frequency Inverse Triple Frequency ($\mathcal{PF}\mathcal{ITF}$). REWOrD also considers paths between concepts to refine relatedness. An experimental evaluation showed that REWOrD obtains results comparable with the state of the art while being more flexible. In fact, it does not require preprocessing of data and can exploit any source of knowledge for which a SPARQL endpoint is available.

An interesting line of future research is the combination of knowledge from different sources. For instance, when comparing *Stanley Kubrick* with *David Lynch* using DBPedia, it would be interesting to exploit links that DBPedia has with LinkedMDB (via `owl:sameAs`) to get information from both sources. Exploiting REWOrD for text relatedness and word sense disambiguation are other interesting challenges.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2008. DBpedia: A Nucleus for a Web of Open Data. In *Proc. of International Semantic Web Conference (ISWC)*, volume 4825 of *LNCS*, 722–735.

Bollegala, D.; Matsuo, Y.; and Ishizuka, M. 2007. Measuring Semantic Similarity between Words Using Web Search Engines. In *Proc. of International World Wide Conference (WWW)*, 757–766.

Budanitsky A, H. G. 2001. Semantic Distance in WordNet: an Experimental Application Oriented Evaluation of Five Measures. In *Proc. of the NAACL Workshop on WordNet and Other Lexical Resources*, 29–34.

Freitas, A.; Oliveira, J. a. G.; O'Riain, S.; Curry, E.; and Da Silva, J. C. P. 2011. Querying Linked Data Using Semantic Relatedness: A Vocabulary Independent Approach. In *Proc. International Conference on Natural Language Processing and Information Systems (NLDB)*, 40–51.

Gabrilovich, E., and Markovitch, S. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proc. of International Join Conference on Artificial Intelligence (IJCAI)*, 1606–1611.

Heath, T., and Bizer, C. 2011. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool.

Landauer, T. K., and Dumais, S. T. 1997. Solution to Plato's problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge. *Psychological Review* (104):211–240.

Mihalcea, R.; Corley, C.; and Strapparava, C. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proc. of National Conference on Artificial Intelligence (AAAI)*, 775–780.

Miller, G., and Charles, W. 1991. Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes* 6:1–28.

Milne, D., and Witten, I. 2008. An Effective, Low-cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *Proc. of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, 25–30.

Oren, E.; Gerke, S.; and Decker, S. 2007. Simple Algorithms for Predicate Suggestions Using Similarity and Co-occurrence. In *Proc. of European Semantic Web Conference (ESWC)*, volume 4519 of *LNCS*, 160–174.

Patwardhan, S.; Banerjee, S.; and Pedersen, T. 2003. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proc. of International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 2588 of *LNCS*, 241–257.

Pedersen, T.; Pakhomov, S. V. S.; Patwardhan, S.; and Chute, C. G. 2007. Measures of Semantic Similarity and Relatedness in the Biomedical Domain. *Journal of Biomedical Informatics* 40(3):288–299.

Pirró, G., and Euzenat, J. 2010. A Feature and Information Theoretic Framework for Semantic Similarity and Relatedness. In *Proc. of International Semantic Web Conference (ISWC)*, volume 6496 of *LNCS*, 615–630.

Pirró, G.; Ruffolo, M.; and Talia, D. 2008. Advanced Semantic Search and Retrieval in a Collaborative Peer-to-Peer System. In *Proc. of International Workshop on Use of P2P, Grid and Agents for the Development of Content Networks (UPGRADE-CN)*, 65–72. ACM.

Ponzetto, S. P., and Strube, M. 2007. Knowledge Derived from Wikipedia for Computing Semantic Relatedness. *Journal of Artificial Intelligence Research* 30:181–212.

Rada, R.; Mili, H.; and Bicknell, E. andBlettner, M. 1989. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics* 19:17–30.

Resnik, P. 1995. Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proc. of International Join Conference on Artificial Intelligence (IJCAI)*, 448–453.

Rubenstein, H., and Goodenough, J. B. 1965. Contextual Correlates of Synonymy. *Communications of the ACM* 8(10):627–633.

Salton, G.; Wong, A.; and Yang, C. S. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM* 18(11):613–620.

Schaeffer, B., and Wallace, R. 1969. Semantic Similarity and the Comparison of Word Meanings. *J. Experiential Psychology* 82:343–346.

Turney, P. D. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proc. of European Conference on Machine Learning (ECML)*, 491–502.

Tversky, A. 1977. Features of Similarity. *Psychological Review* 84(2):327–352.

Yeh, E.; Ramage, D.; Manning, C. D.; Agirre, E.; and Soroa, A. 2009. Wikiwalk: Random walks on Wikipedia for Semantic Relatedness. In *Proc. of Workshop on Graph-based Methods for Natural Language Processing*, 41–49.