Computing Semantic Similarity Using Ontologies

Rajesh Thiagarajan, Geetha Manjunath, and Markus Stumptner
HP Laboratories
HPL-2008-87

Keyword(s):
semantic web, semantics, matching, similarity, ontology, user profiles, spreading activation networks

Abstract:
Determining semantic similarity of two sets of words that describe two entities is an important problem in web mining (search and recommendation systems), targeted advertisement and domains that need semantic content matching. Traditional Information Retrieval approaches, even when extended to include semantics by performing the similarity comparison on concepts instead of words/terms, may not always determine the right matches when there is no direct overlap in the exact concepts that represent the semantics. As the entity descriptions are treated as self-contained units, the relationships that are not explicit in the entity descriptions are usually ignored. We extend this notion of semantic similarity to consider inherent relationships between concepts using ontologies. We propose simple metrics for computing semantic similarity using spreading activation networks with multiple mechanisms for activation (set based spreading and graph based spreading) and concept matching (using bipartite graphs). We evaluate these metrics in the context of matching two user profiles to determine overlapping interests between users. Our similarity computation results show an improvement in accuracy over other approaches, when compared with human-computed similarity. Although the techniques presented here are used to compute similarity between two user profiles, these are applicable to any content matching scenario.

External Posting Date: July 6, 2008 [Fulltext] Approved for External Publication

Internal Posting Date: July 6, 2008 [Fulltext]

Submitted to ISWC 08, the International Semantic Web Conference (ISWC), 2008, Karlsruhe, Germany

© Copyright 2008 Hewlett-Packard Development Company, L.P.

Computing Semantic Similarity Using Ontologies

Rajesh Thiagarajan[1], Geetha Manjunath[2], and Markus Stumptner[1]

[1] Advanced Computing Research Centre, University of South Australia
{cisrkt|mst}@cs.unisa.edu.au
[2] Hewlett-Packard Labs, Bangalore, India
geetha.manjunath@hp.com

Abstract. Determining semantic similarity of two sets of words that describe two entities is an important problem in web mining (search and recommendation systems), targeted advertisement and domains that need semantic content matching. Traditional Information Retrieval approaches, even when extended to include semantics by performing the similarity comparison on concepts instead of words/terms, may not always determine the right matches when there is no direct overlap in the exact concepts that represent the semantics. As the entity descriptions are treated as self-contained units, the relationships that are not explicit in the entity descriptions are usually ignored. We extend this notion of semantic similarity to consider inherent relationships between concepts using ontologies. We propose simple metrics for computing semantic similarity using spreading activation networks with multiple mechanisms for activation (set based spreading and graph based spreading) and concept matching (using bipartite graphs). We evaluate these metrics in the context of matching two user profiles to determine overlapping interests between users. Our similarity computation results show an improvement in accuracy over other approaches, when compared with human-computed similarity. Although the techniques presented here are used to compute similarity between two user profiles, these are applicable to any content matching scenario.

Key words: semantic web,semantics,matching,similarity,ontology,user profiles,spreading activation networks

1 Introduction

Similarity denotes the relatedness of two entities. Determining similarity of two entities is an important problem in the domain of web mining (search, recommendation systems), targeted advertising (matching description of the user with the keywords describing the target audience of the advertisement) and so on. It is common in Information Retrieval (IR) frameworks to represent the entities such as documents or queries in the so-called *bag-of-words* (BOW) format. A BOW format is a set of weighted terms that best describe the entity so that the similarity between two entities can now be computed using just their BOW representations.

A number of similarity measurement techniques such as the cosine similarity measure [9, 3], Dice's coefficient [13] and Jaccard's index [6] have been defined to compute this similarity. Among these, the most widely applied similarity

measure, the cosine similarity measure [9] has been applied to content matching scenarios such as document matching [9], ontology mapping [10], document clustering [11], multimedia search [2], and as a part of web service matchmaking frameworks [7, 4, 15].

Motivation: Even though the term vector similarity matching is used in a number of such applications for its simplicity and reasonable accuracy, it is well known that considering just the terms results in matching problems due to lack of semantics in the representation. Problems due to polysemy (terms such as *apple, jaguar* having two different meanings) and synonymy (two words meaning almost the same thing such as *glad* and *happy*) can be solved if entities are described using concepts instead of terms. However, these approaches may not still determine the right matches when there is no direct overlap in the exact concepts that represent the semantics, as they treat the entity descriptions as self-contained units. For example, do two users with *Yahoo* and *Google* in their respective profiles have nothing in common? There does seem to be an intersection in these users' interests for Web-based IT companies or web search tools! Such overlaps are missed as current approaches work under the assumption that the BOW representations contain all the information about the entities. As a result, relationships that are not explicit in the representations are usually ignored. Furthermore, these mechanisms cannot handle entity descriptions that are at different levels of granularity or abstractions (Eg: *jazz* and *music*) as the implicit relationship between the concepts is ignored.

The Role of the Semantic Web: The web is growing to be a source of domain intelligence with human knowledge about specific domains captured in the form of ontologies such as Wordnet, Cyc, ODP, Wikipedia and many more available on Swoogle. These ontologies capture semantic relationships between concepts or vocabulary used in a particular domain and can potentially be used to discover inherent relationships between descriptions of entities. One of the critical factors that hindered the adoption of such ontologies so far was the absence of a machine-readable, interchangeable representation of this knowledge that these ontologies offer. The Semantic Web [1] technology with its core Resource Description Framework (RDF)[3] provides this much needed representation formalism. The ontologies mentioned above are now available in RDF and can therefore be adopted to discover the inherent relationships between descriptions of entities to address the challenged outlined earlier.

Our Contribution: In this paper, we extend the notion of semantic similarity between two entities to consider inherent relationships between concepts/words appearing in their respective BOW representation, through a process called *spreading*. Spreading, the process of including additional related terms to an entity description by referring to an ontology such as Wordnet and Wikipedia, has been used in earlier frameworks ([12, 14, 10, 2]). We build on such earlier techniques and propose simple metrics for computing similarity with ontology-based spreading activation networks. We evaluate multiple mechanisms for activation (set based spreading and graph based spreading) and concept matching (set intersection and use of bipartite graphs)in the context of matching two user profiles to determine overlapping interests between users. Our similarity computation results show an improvement in accuracy over other approaches, when compared

[3] http://www.w3.org/RDF/

with human-computed similarity. Although the techniques presented here are used to compute similarity between two user profiles, these are applicable to any content matching scenario.

Structure of this document: In the next section, we provide a brief background to currently used techniques and describe related work in this area. We detail our new approach in the subsequent sections. The process of spreading as a means to consider the inherent relationships that might exist between two entity descriptions is described in Section 3.1. In Section 3.2, we describe a procedure to incrementally extend a given entity description with related terms so that conventional means of computing similarity can be employed. The subsequent section describes a mechanism of constructing spreading activation networks using ontologies and explains an optimal term/concept matching technique in bipartite graphs to determine similarity (Section 3.2). We describe our evaluation procedure for the user interest matching scenario in Section 4 and share our improved results. Finally, we summarize our contributions and state our future work in Section 5.

2 Background and Related Work

We briefly revisit the terms defined in the previous section to provide their formal definition as a background of the next few sections. A BOW format referred to earlier, where an entity is represented by a set of pairs, is denoted as $\langle t_i, w_i \rangle$ where t_i is a term that describes the entity and $t_i \in terms(E)$. w_i is the weight[4] of the term that denotes the importance of the respective term in describing the entity. The BOW format simplifies the problem of computing similarity between the entities to computing the similarity between their BOW representation.

The popular cosine similarity measure or the term vector similarity between any two entities is the cosine angle between two vectors modeled out of the BOW representation of the two entities. If the vector representation of an entity e_j is $\overrightarrow{V}(e_j)$ and the *Euclidean length* ($|\overrightarrow{V}(e_j)|$) of an entity e_j is $\sqrt{\sum_{i=1}^{n} w_i^2}$, the similarity of the entities e_j and e_k is

$$sim_{cos}(e_j, e_k) = \frac{\overrightarrow{V}(e_j) \cdot \overrightarrow{V}(e_k)}{|\overrightarrow{V}(e_j)||\overrightarrow{V}(e_k)|} \quad (1)$$

A recent study [5] extends this cosine similarity metric to include semantics by performing a cosine similarity on ontology concepts that describe an entity instead of words/terms. The representation they use is very similar to BOW except that a semantic mapping mechanism maps every term to semantic concept. This new description of the entities is called the *bag of concepts* (BOC) representation[5]. The ontology used in [5] is Wikipedia. Every Wikipedia page title is considered an ontology concept. The study shows that pre-processing the documents into this BOC format prior to computing the cosine similarity is more accurate than the term vector similarity measure. As mentioned earlier,

[4] In the document matching scenario, w_i is usually computed as a product of Term Frequency (TF) and Inverse Document Frequency (IDF) [9].

[5] t_i is an ontology concept

this approach will not still determine a semantic inexact match when there is no direct overlap in the concepts that represent the semantics.

A user preference learning mechanism that drives a personalized multimedia search is presented in [2]. The learning process utilizes ontologies as a means to comprehend user interests and establishes the need to consider related concepts to improve search quality. Our work on spreading builds on the notion of considering related concepts. While their results suggest that personalized search is of better quality in comparison to normal search, it is not conclusive whether the consideration of related terms contributes to these improvements. On the other hand, we show that our spreading process indeed improves the accuracy of our new similarity measures.

In [10], an ontology concept is represented using a profile with words describing the concept. A propagation technique to enrich the profile with words from neighboring concepts is also presented. Such profiles are subsequently used to determine closeness (using cosine metric) between concepts they represent. The problem of determining the closeness between two ontology concepts is reduced to a sub-graph matching problem in [16] where a recursive procedure to match edges and nodes of the sub-graphs is proposed. While both [10] and [16] are only able to determine closeness between two concepts (or words), we present several measures to compute similarity between two weighted sets of concepts (or words).

A word sense disambiguation technique that determines the right sense of words in a sentence by activating a semantic network constructed by referring to a thesaurus [6] is presented in [12]. The evaluation demonstrates the effectiveness of utilizing an ontology to disambiguate word senses. One of our similarity measures that processes the semantic network post spreading the BOWs builds on this earlier work. Our work differs from this earlier work in the treatment of the results of the activation process. While the previous work utilizes the results of the activation to map a meaning to every word, our work maps an aggregate of the activation results to a similarity value.

3 Our Solution

In this section, we describe the complete details of our approach to compute semantic similarity using ontologies. We show two techniques to compute and represent the extended entity descriptions derived from an ontology and the variant metrics that we can use with them.

3.1 Spreading to Capture Semantics in Entity Descriptions

Spreading is the process of including the terms that are related to the original terms in an entity's description (ED; either BOW or BOC) by referring to an ontology. Previous studies have shown that the spreading process improves accuracy and overcomes the challenges caused by inherent relationships and

[6] Wordnet is used for evaluation

Polysemy[7] in a number of other frameworks such as word sense disambiguation process [12, 14], ontology mapping [10], and personalized multimedia access [2]. We use this spreading process to facilitate the semantic similarity computation process.

Let us study the earlier mentioned simple example of two users having *google* and *yahoo* in their profile in detail to understand the spreading process better.

Example 1. Consider computing the similarity of the following EDs

- $e_1 = \{\langle google, 1.0\rangle\}$, and
- $e_2 = \{\langle yahoo, 2.0\rangle\}$.

A simple intersection check between the EDs results in an empty set (i.e. $e_1 \cap e_2 = \emptyset$) indicating their un-relatedness (cosine similarity is 0). However, if we were to manually judge the similarity of these two entities we would give the similarity of the entities a value greater than 0. This is because we judge the similarity not just by considering the two terms from the EDs but also by considering the relationships that might exist between them. We are able to establish the fact that both *google* and *yahoo* are search engine providers.

Now let us see the effectiveness of spreading in the similarity computation process in the same example.

Spreading the EDs e_1 and e_2, by referring to Wikipedia parent category relationship, extends the EDs to

- $e'_1 = \{\langle google, 1.0\rangle, \langle internet\ search\ engines, 0.5\rangle\}$, and
- $e'_2 = \{\langle yahoo, 2.0\rangle, \langle internet\ search\ engines, 1.0\rangle\}$.

The simple intersection check results in a non-empty set (i.e. $e'_1 \cap e'_2 \neq \emptyset$) indicating their relatedness (cosine similarity is 0.2). The result of the spreading (i.e. the inclusion of the related term *internet search engines*) process makes sure that any relationship that exists between the EDs are taken into consideration.

How does one determine the weights of the new terms introduced? How manu such new terms can we go on introducing? Typically an ontology $\mathcal{O}$ holds knowledge about terms/concepts and their relationship with other terms/concepts. Given a term t_i, the spreading process utilises $\mathcal{O}$ to determine the terms that are related to t_i (denoted as $related_{\mathcal{O}}(t_i)$). Although spreading the profiles with all the related terms allows for a comprehensive computation to be carried out, in practice the addition of all the related terms leads to the dilution of the profiles with noise or unrelated terms. This dilution may have negative implications on the computation process where the similarity in the noise may contribute to the similarity values between entities. It is therefore desirable to have control over the types of relationships to be considered during this spreading process.

We use two main parameters to control the spreading process: relationship types and weight functions. For example, spreading based on Wikipedia may be limited to only spreading only along the parent categories. Additionally, a set of weight functions defines the weights for each of the allowed relationship type.

[7] Note that the difference in the semantics may also be captured by the mapping that exists between a term and an ontological concept. For example, if the term 'apple' is mapped to a computing ontology then it probably means the company and not the fruit.

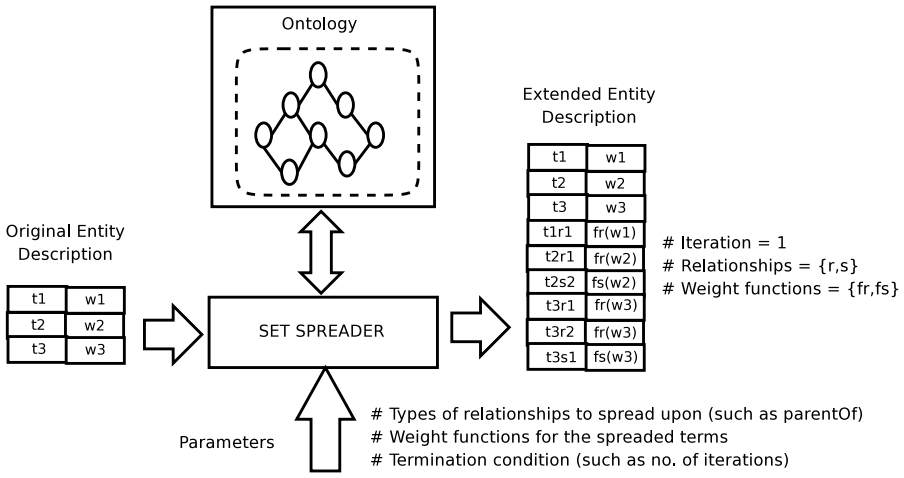


Fig. 1: Set Spreading Process

The weights of the related terms are proportional to the weights of their related original term because the weight w_i of a term t_i indicates the importance of the term within a profile. The rationale behind this proportionality is that the weights of the terms related to an original terms with higher weight should be higher than weights of the terms related to an original term with lower weight. Moreover, the weights may differ according to the semantics of the relationships allowed. For example, a spreading process based on Wordnet limited to types synonym and antonym can have functions $t_{ij} = w_i \times 0.9$ and $t_{ij} = w_i \times -0.9$ respectively.Therefore, the weights of a related node t_{ij} is a function of the w_i and the type of the relationship.

Inspired by [5, 12], we propose two schemes for representing the related terms post-spreading: extended set and semantic network. The two schemes are conceptually similar they only differ in the implementation where one process returns an extended set and the other returns a graph. However, different similarity metrics based on the edges and paths in the graph can be established in the latter case as seen later.

Set Spreading: Set spreading is the process of extending an ED such that the related terms, which are determined with respect to an ontology, are appended to the original set of terms. Figure 1 shows the set spreading process. For an ontology $\mathcal{O}$, set spreading an ED E results in E' such that the set of terms $terms(E') = \{t_1, \ldots, t_n, t_{11}, \ldots, t_{nm}\}, terms(E) \subseteq terms(E')$ where $\forall t_{ij} | t_{ij} \in related_{\mathcal{O}}(t_i)$ and there exist a path from t_i to t_j.

Set spreading is an iterative process. After each iteration, the related terms of the terms from the previous iterations are appended to the ED. Two parameters control the termination of the iterative set spreading process: number of iterations and exhausted relationships. The spreading process is terminated if there are no related terms to spread the ED with. That is, the spreading process is stopped at the iteration in which $\forall t_i \in terms(E) | related_{\mathcal{O}}(t_i) = \emptyset$. Alternatively, an arbitrary number can be set by the user to limit the number of iterations.

Graph Spreading: Graph spreading is the process where terms from two EDs and the related terms are build into a (related-term) graph representation that we call as a semantic network. This network is then used to establish the similarity between the entities.The goal is to establish a semantic connectivity within this graph between the terms from two EDs. Figure 2 shows the graph spreading process.

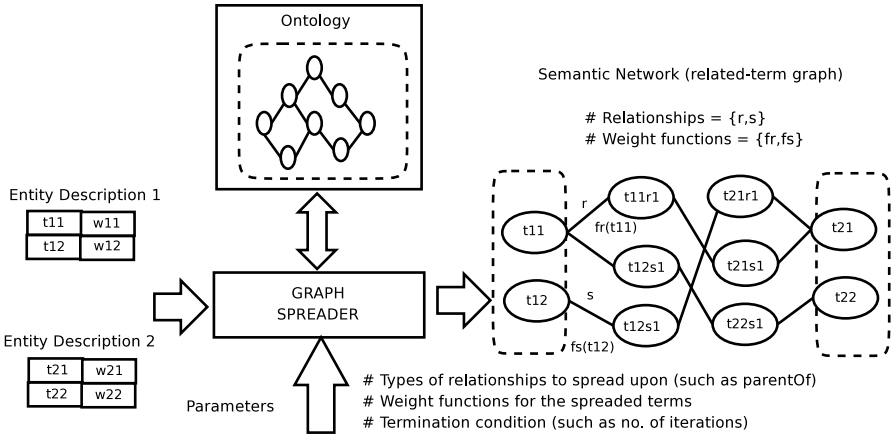


Fig. 2: Graph Spreading Process

The spreading process begins with an empty graph. And proceeds as described in the following.

STEP 1 The terms within the two EDs t_i are added as nodes in the graph.
STEP 2 The terms t_i are added into a list called OPEN.
STEP 3 For each term t_i in the OPEN, the related terms t_{ij} are determined.
STEP 4 The terms t_{ij} are added as nodes in the graph. An edge connecting every t_{ij} from t_i is added to the graph.
STEP 5 The terms t_i are removed from OPEN and the terms t_{ij} are added to OPEN.
STEP 6 If at-least one of the termination conditions is met then the process is terminated and the graph is returned. Otherwise continue with STEP 3.

Unlike the set spreading process where the relationship between a term in an ED and its related term is not preserved, the graph spreading process preserves this in the form of a graph edge. This allows for the development of similarity measures where various methods for handling the related terms based the semantics of the relationships can be utilized on the same network.

In the graph spreading, apart from *Number of Iterations* and *Exhausted Relationships*, the spreading process is terminated if there exists a path between every pair of the term nodes from the two EDs. This condition best suits the ontologies that have a top root element which subsumes the rest of the elements in the ontology. For example, Wordnet based spreading can be tuned to employ this termination condition when path from individual terms to the root suffices to terminate the spreading. In less rigorous ontologies such as Wikipedia the

category graph may be not be able to support this condition as there may not be a single root. In such a case, the spreading process is terminated if there exists at least one path from every node that belongs to the smallest of the two EDs to the nodes in the other ED. This condition is less rigorous than the previous one.

3.2 Similarity Computation

Our set spreading process enriches the EDs by appending the related terms in order to capture all the relationships between the EDs. So for the set based spreading the same cosine similarity technique defined in Equation 1 is applicable to compute similarity between the extended BOWs or BOCs. Our iterative similarity computation procedure based on set spreading begins with the measuring similarity of the original EDs, incrementally extends the EDs until termination while computing the similarity between EDs at every iteration. The procedure is as follows.

STEP 1: Compute similarity between original two EDs.
STEP 2: Spread the two EDs if none of the termination conditions are not met else go to STEP 5.
STEP 3: Compute cosine similarity (Equation 1) between the two extended EDs.
STEP 4: Go to STEP 2.
STEP 5: Compute the mean of the similarity values computed in all the iterations (see Section 4.4 for the rationale behind this step).
STEP 6: Return the mean similarity value.

Similarity Computation: Graph-based In this section, we present our similarity measurement techniques that process the semantic network, constructed from the related terms as per the process outlined in Figure 2, to compute similarity between two entities. A snapshot of the semantic network construction process is shown in Figure 3.

Following the construction of the semantic network the similarity values are computed by either reducing the graph to a bipartite graph or by activating the graph (treated as a Spreading Activation Network (SAN)) with an activation strategy. We have implemented both these techniques for evaluation.

Similarity Computation by Matching Bipartite Graph By omitting the intermediate related nodes and considering only the path length between the nodes representing the original ED terms, the semantic network can be considered to a bipartite graph (shown on the left side of Figure 4). The nodes of the first ED and second ED are the two vertex sets of the bipartite graph where the edge denotes the length between the original term nodes as obtained from the semantic network. Once the bipartite graph is derived, we are able to apply standard algorithms for optimal matching of the bipartite graph. Our similarity measures based on optimal bipartite matching operates under the simple notion that the nodes with higher weights and that are closely located contribute more to the similarity of the entities and viceversa.

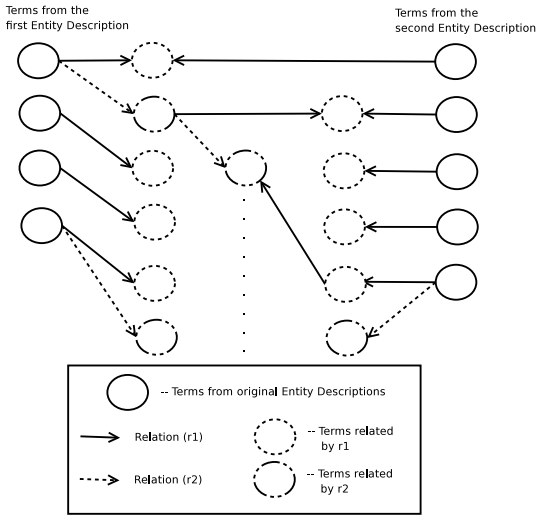


Fig. 3: A Snapshot of the Related-Term Graph Building Process

Each node v_i^{ed} in the semantic network is a pair $\langle t_i, w_i \rangle$ where $ed = 1$ or 2 denoting which ED term the node denotes. The $path(v_i^1, v_j^2)$ denotes the set of edges between two nodes v_i^1 and v_j^2 in the semantic network. All the edges between any two nodes with different terms in the semantic network have uniform weights $\forall e \in path(v_i^1, v_j^2)$ set $wt(e) = 1$. For any two vertices v_i^1 and v_j^2 the distance between the them is

$$len(v_i^1, v_j^2) = \begin{cases} 0, & \text{if } t_i = t_j \\ \sum_{\forall e_k \in path(v_i^1, v_j^2)} wt(e_k), & \text{otherwise} \end{cases}$$

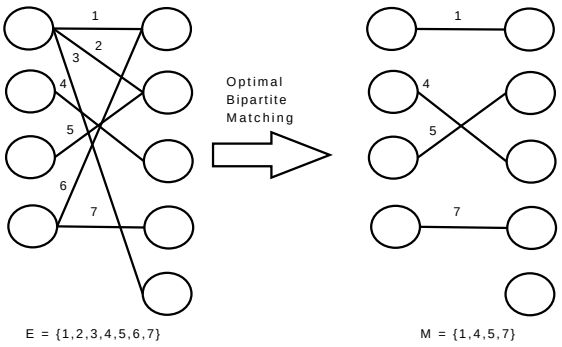


Fig. 4: Matching the Bipartite Graph (Hungarian Algorithm)

The bipartite graph representation G of the EDs ed_1 and ed_2 is a pair $G = \langle V, E \rangle$ where

- $V = V^1 \cup V^2$ where V^1 denotes the vertices from the first ED ed_1 and V^2 denotes the vertices from the second ED ed_2.

- $V^1 = \{v_1^1, v_2^1, \ldots, v_n^1\}$ and $V^2 = \{v_1^2, v_2^2, \ldots, v_m^2\}$ where $n \leq m$ and $v_i^k = \langle t_i^k, w_i^k \rangle$ is a term.
- $E = \{e_{11}, e_{12}, \ldots, e_{ij}\}$ where $i = \{1, 2, \ldots, n\}$, $j = \{1, 2, \ldots, m\}$ and $len(v_i^1, v_j^2)$ denotes the path length between then vertices v_i^1 and v_j^2.

Given the bipartite representation G, the optimal matching $E' \subseteq E$ between two vertex sets is computed using the Hungarian Algorithm [8].The optimal bipartite graph (shown on the right side of Figure 4) is $G' = \langle V, E' \rangle$ where $E' \subseteq E$ such that $\sum_{\forall e_{ij} \in E'} len(v_i^1, v_j^2)$ is optimal. Given the weights of vertices in the representation $W^{12} = w_i^1 \cup w_j^2$, these are normalized (value [0-1]) to $W^{12'} = w_i^{1'} \cup w_j^{2'} = \{w_1^{1'}, \ldots, w_i^{1'}, w_1^{2'}, \ldots, w_j^{2'}\}$ where $\forall_{w_l^k \in W^{12'}}$

$$w_l^{k'} = \frac{w_l^k}{max(W^{12'})}$$

Aggregate Path Distances: Abiding by our notion that the closer nodes with higher weights contribute more to the similarity value, we present three (slightly different) path length aggregation measures for empirical evaluation. The paths distance of an edge e_{ij} in the optimal bipartite graph is defined as

$$path(e_{ij}) = \begin{cases} 1, & \text{if } len(v_i^1, v_j^2) \text{ is } 0 \\ 0, & \text{if } len(v_i^1, v_j^2) \text{ is } \infty \\ \frac{w_i^{1'} \times w_j^{2'}}{len(v_i^1, v_j^2)}, & \text{otherwise} \end{cases}$$

The Euler path distance of an edge e_{ij} in the optimal bipartite graph is defined as

$$eupath(e_{ij}) = \begin{cases} 1, & \text{if } len(v_i^1, v_j^2) \text{ is } 0 \\ 0, & \text{if } len(v_i^1, v_j^2) \text{ is } \infty \\ \frac{w_i^{1'} \times w_j^{2'}}{e^{len(v_i^1, v_j^2)}}, & \text{otherwise} \end{cases}$$

The Euler half path distance of an edge e_{ij} in the optimal bipartite graph is defined as

$$euhalf(e_{ij}) = \begin{cases} 1, & \text{if } len(v_i^1, v_j^2) \text{ is } 0 \\ 0, & \text{if } len(v_i^1, v_j^2) \text{ is } \infty \\ \frac{w_i^{1'} \times w_j^{2'}}{e^{\left(\frac{len(v_i^1, v_j^2)}{2}\right)}}, & \text{otherwise} \end{cases}$$

The aggregate distance of all the matching edges of the bipartite graph is given by the sum of their path distances.

Similarity Measures Given two entities ed_1 and ed_2, the similarity of between them using aggregate paths distance in the optimal bipartite graph is

(2)

$$sim_{path}(ed_1, ed_2) = \frac{\sum_{\forall e_{ij} \in E'} path(e_{ij})}{min(size(terms(ed_1)), size(terms(ed_2))) \times max(path(e_{ij}))}$$

The similarity of the two entities using aggregate Euler paths distance in the optimal bipartite graph is defined as
(3)

$$sim_{eupath}(ed_1, ed_2) = \frac{\sum_{\forall e_{ij} \in E'} eupath(e_{ij})}{min(size(terms(ed_1)), size(terms(ed_2))) \times max(eupath(e_{ij}))}$$

The similarity of the two entities using aggregate Euler paths half distance in the optimal bipartite graph is defined as
(4)

$$sim_{euhalf}(ed_1, ed_2) = \frac{\sum_{\forall e_{ij} \in E'} euhalf(e_{ij})}{min(size(terms(ed_1)), size(terms(ed_2))) \times max(euhalf(e_{ij}))}$$

Similarity Computation using Activation Values This similarity computation metric is inspired by the work presented in [12]. While the earlier work proposed for word sense disambiguation using Wordnet as a means to spread a SAN (snapshot shown the Figure 3), we use a similar activation strategy to compute similarity between two entities. A brief introduction to the activation process is presented in the following. For a more detailed discussion the reader is pointed to [12]. The overall improvement of this technique over the one described in previous section is when two neighbours influence the activation value at a specific node, in which case, the activation value is aggregated. However, this technique is computationally intensive.

The SAN activation process is iterative. $A_j(p)$ denotes the activation value of node j at iteration p. All the original term nodes take their term weights as their initial activation value $A_j(0) = w_j$. The activation value of all the other nodes are initalized to 0.
In each iteration,

- Every node propagates its activation to its neighbours.
- The propagated value is a function of the nodes current activation value and weight of the edge (see [12]) that connects them (denoted as $O_j(p)$).

After a certain number of iterations, the highest activation value among the nodes that are associated with each of the original term node is retrieved into a set $ACT = \{act_1, act_2, ..., act_{n+m}\}$. The aggregate of the these activation values can be mapped to the similarity between entities under the intuition that the nodes with higher activation values are typically the ones that have value contributions from both the entities and hence should contribute more to similarity and viceversa.

Therefore, the similarity value is the sum of the set ACT normalized to a value between 0 and 1. The SAN-based similarity between two EDs ed_1 and ed_2 is

$$sim_{san}(ed_1, ed_2) = \frac{\sum_{\forall act_i \in ACT} act_i}{|ACT| \times max(act_i)} \quad (5)$$

4 Evaluation and Results

We use a user profile matching scenario as a platform to evaluate the similarity measurement techniques presented in this paper.

4.1 User Profile Matching

Profiles are generic structures used to capture characteristics of entities. Here user profiles capture the characteristics/interests of users. We consider the computation of semantic similarity between two user profiles in order to determine the overlapping interests between two users. We use a custom-built software called profile builder to generate user profiles. The user profiles are generated by analysing the documents (such as web pages visited by the user) that belong to the user. Both the BOW (word profiles) and BOC (terms are Wikipedia concepts; Wiki profiles) representation of the user interests are generated by the profile builder software.

The overlapping interests between two users is denoted by the similarity measure between the two user profiles. Although the approaches presented here are used to compute similarity between two user profiles, the techniques implemented are applicable to any content matching scenario because profiles are generic structures to capture characteristics of entities (users in this case).

4.2 User Study

Measure	**Description**
Base	The similarity measure derived from the user judgements
COS-Word	Cosine similarity measure between words in word profiles (Equation 1)
COS-Con	Cosine similarity measure between Wiki concepts in Wiki profiles (Equation 1)
COS-5n	Mean cosine similarity measure between Wiki concepts in Wiki profiles after 5 iterations of set spreading
COS-10n	Mean cosine similarity measure between Wiki concepts in Wiki profiles after 10 iterations of set spreading
Bi-PATH	Similarity measure after graph spreading as defined in Equation 2
Bi-EU	Similarity measure after graph spreading as defined in Equation 3
Bi-EUby2	Similarity measure after graph spreading as defined in Equation 4
SAN	Similarity measure after graph spreading as defined in Equation 5

Table 1: Glossary of the Similarity Measures

A pilot study was conducted as a part of the evaluation process. The study conducted had 10 participants (Labelled A-J). From each of the participants, 5 to 10 documents that in the participant's opinion best describe their research were collected. Along with the documents, the participants were asked to give 5 keywords for each of their document that in their opinion best described the document. An aggregated set of keywords for each of the participants was derived from the document keywords that were suggested by the participants. Therefore, all the participants were represented by the derived set of keywords which is referred to as the participant's profile.

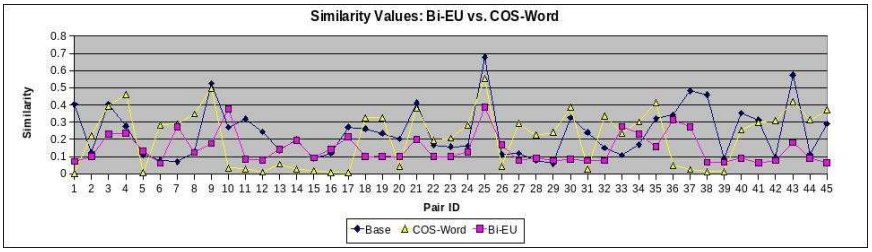
Based on these profiles the participants provided two similarity judgements as described in the following. Each of the participant judged the similarity between their profile and other profiles. Additionally, each of the participant judged the similarity between every pair of profiles. The mean of the subjective judgements

provided by the participants were used as the base/reality values to evaluate our similarity measures presented in Table 1.

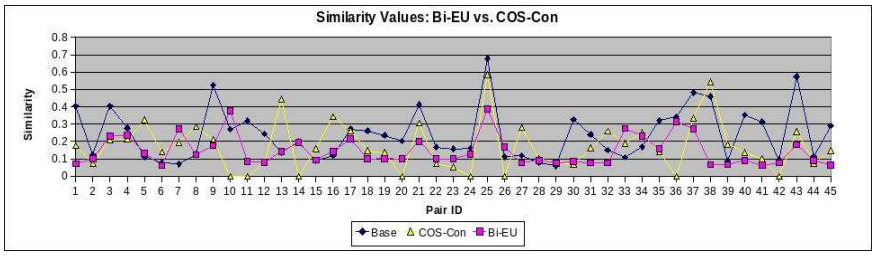
The participants judging the similarity between profiles were given a scale of [0–100] with 100 being the maximum similarity to denote their judgement. These judgements were then normalized to a value between [0–1]. In addition, the judgements were clustered into slabs[8] to tolerate marginal errors/differences in the similarity judgements.

4.3 Experiments and Analysis

Using the profile builder, the word and wiki profiles were generated for each of the participant based on their documents. All the approaches listed in Table 1 were allowed to compute the similarity between the profiles.



(a) Similarity Values from the BI-EU Approach with Base and COS-Word



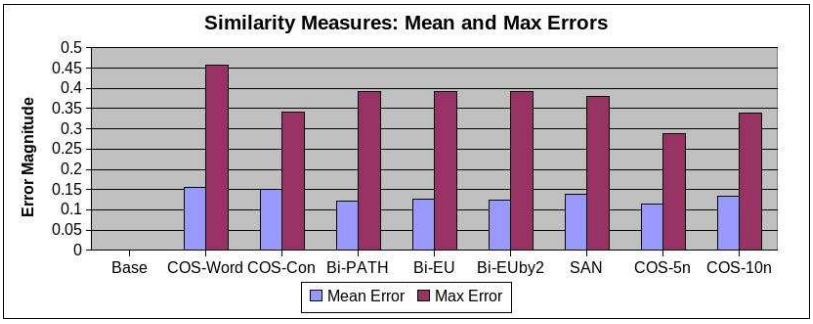
(b) Similarity Values from the BI-EU Approach with Base and COS-Con

Fig. 5: Analysis of the Experimental Results - Part 1

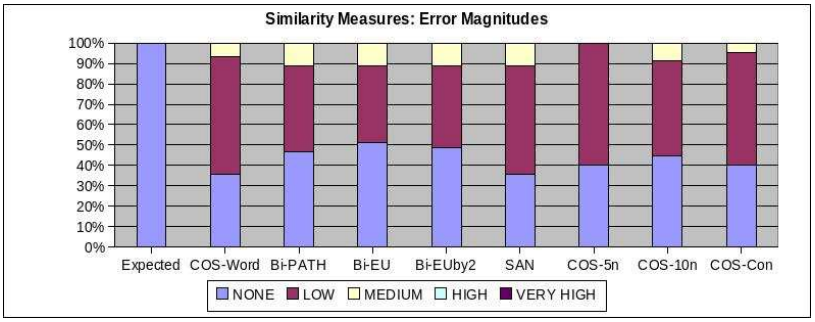
The similarity values for each of the 45 unique pairs of profiles was computed by all of the approaches from Table 1 some results are presented in Figure 5. Cosine similarity or term vector similarity on the BOW representation of documents is a widely used document similarity metric. However, by considering the similarity values in Figure 5a, it is clear that the COS-Word metric is not accurate to compute profile similarity. For a number profile pairs $\{1, 5, 12, \ldots, 38, 39\}$ the COS-Word metric returns 0 as the similarity value. The reason for this behaviour is the absence of common words in these user profiles.

The COS-Con metric in our framework measures the concept vector similarity between profiles. As shown in Figure 5b, the accuracy of the COS-Con

[8] VERY LOW [0–20], LOW [21–40], MEDIUM [41–60], HIGH [61–80], and VERY HIGH [81–100]



(a) Mean and Max Errors of the Approaches



(b) Error Magnitudes of the Approaches

Fig. 6: Analysis of the Experimental Results - Part 2

is better than that of the COS-Word because the concepts in the wiki profiles already capture some semantics that are not captured in the word profiles. Although the accuracy of COS-Con is better in comparison with COS-Word, there are a number of instances $\{10, 11, 14, 20, 24, 26, 36, 42\}$ where the absence of intersecting concepts return 0 as similarity values. The reason for this behaviour is that the computation process does not consider the inherent relationships between the profiles.

By analysing the similarity values computed by all the approaches listed in Table 1 we are able to conclude that our extensions based on set-spreading (COS-5n and COS-10n) and graph-spreading (Bi-PATH, Bi-EU, Bi-EUby2, and SAN) consistently return more accurate results in comparison with COS-Word and COS-Con.

It is clear by observing Figure 6a that the average and maximum error[9] of the COS-Word approach is significantly higher than that of all our approaches. Among our approaches the measures based on Bipartite graphs Bi-PATH, Bi-EU, and Bi-EUby2, and COS-5n have the least average error. In terms of maximum error, the set spreading based approaches COS-5n and COS-10n have the least maximum errors. Analysis of the error magnitudes[10], as shown in Figure 6b, show that COS-Word has the least accuracy because of the low number of er-

[9] Offset from the expected value i.e. $error = |observed - expected|$

[10] Difference in slabs, for example expected = VERY HIGH, observed = VERY LOW results in VERY HIGH error magnitude

rors with magnitude 0. While Bi-EU is the best among all the graph spreading approaches, COS-5n is the best of all the approaches. It is clear from Figures 5 and 6a that the accuracy of the COS-Word approach is the lowest among the measures. Therefore, in general it can be stated that our spreading based computations yield more accurate similarity judgements than the simple vector based counterparts. All of our approaches exhibit improvements accuracy in comparison with COS-Word and COS-Con. In particular, Bi-EU and COS-5n yield more accurate similarity measures between user profiles.

4.4 Monotonicity

The set spreading process described in Section 3.1 extends the terms in the profiles with related terms. Set spreading is an iterative process. New related terms are appended to the original profile sets at each iteration. *Does this mean that the similarity value either increases or does not decrease with the number of iterations?*

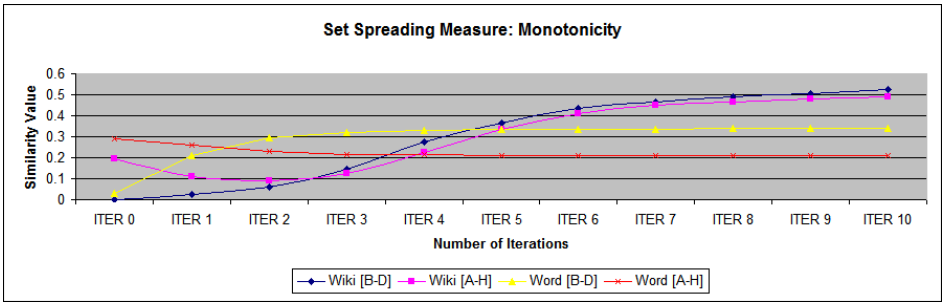


Fig. 7: Monotonocity of the Approaches

As shown in Figure 7, for certain profile pairs such as [B,D] the computation process is monotonic whereas for certain other profile pairs such as [A,H] the process is non-monotonic. This behaviours is consistent across both the ontologies used in our experiments Wikipedia and Wordnet. The reason for non-monotonicity being expressed by certain profile pairs is that the inclusion of related but disjoint terms to the profile results in reduced normalized weights[11]. As a result, the cosine similarity values also decrease. However, once the inclusion of related common terms begins at some iteration the computation process turns monotonic thereafter. For example, in Figure 7 for profile pair [A,H] after iteration 3 the computation process turns monotonic.

Since the set spreading-based similarity computation does not consistently exhibit monotonicity, from our experiments we conclude that the mean of the similarity values computed in every iteration is the more accurate[12] measure of similarity between user profiles.

[11] The length of the profiles increase. Therefore the normalized weights are lesser than the original term weights.

[12] in comparison with determining an iteration number where the similarity values are accurate

5 Conclusion and Future Work

We presented a number of similarity computation measures that utilises spreading as a means to capture the semantics of the description of entities. The evaluation of the similarity measures shows the improvements in accuracy that is achieved over existing traditional similarity computation methods. Further work in this direction include considering the hierarchical information from the profile builder in the similarity computation process and to experiment the measures with other ontologies such as the ODP. We are also exploring techniques to automatically determine the ontology to be used for spreading using a public ontology repository like Swoogle.

References

1. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.
2. Iván Cantador, Miriam Fernández, David Vallet, Pablo Castells, Jérôme Picault, and Myriam Ribière. A multi-purpose ontology-based approach for personalised content filtering and retrieval. In *Advances in Semantic Media Adaptation and Personalization*, pages 25–51. 2008.
3. Devanshu Dhyani, Wee Keong Ng, and Sourav S. Bhowmick. A survey of web metrics. *ACM Comput. Surv.*, 34(4):469–503, 2002.
4. Xin Dong, Alon Y. Halevy, Jayant Madhavan, Ema Nemes, and Jun Zhang. Similarity search for web services. In *Proc. of VLDB*, pages 372–383, 2004.
5. Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. IJCAI*, 2007.
6. Lieve Hamers, Yves Hemeryck, Guido Herweyers, Marc Janssen, Hans Keters, Ronald Rousseau, and André Vanhoutte. Similarity measures in scientometric research: The jaccard index versus salton's cosine formula. *Inf. Process. Manage.*, 25(3):315–318, 1989.
7. Matthias Klusch, Benedikt Fries, and Katia P. Sycara. Automated semantic web service discovery with owls-mx. In *In Proc. AAMAS*, pages 915–922, 2006.
8. Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
9. Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
10. Ming Mao. Ontology mapping: An information retrieval and interactive activation network based approach. In *Proc. ISWC*, 2007.
11. Horacio Saggion, Adam Funk, Diana Maynard, and Kalina Bontcheva. Ontology-based information extraction for business intelligence. In *Proc. ISWC*, 2007.
12. George Tsatsaronis, Michalis Vazirgiannis, and Ion Androutsopoulos. Word sense disambiguation with spreading activation networks generated from thesauri. In *Proc. IJCAI*, 2007.
13. C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.
14. Jean Véronis and Nancy Ide. Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries. In *In Proc. COLING*, pages 389–394, 1990.
15. Yiqiao Wang and Eleni Stroulia. Semantic structure matching for assessing web-service similarity. In *Proc. ICSOC 2003*, pages 194–207, 2003.
16. Haiping Zhu, Jiwei Zhong, Jianming Li, and Yong Yu. An approach for semantic search by matching rdf graphs. In *In Proc. FLAIRS*, pages 450–454, 2002.