OSS: A Semantic Similarity Function based on Hierarchical Ontologies

Vincent Schickel-Zuber and Boi Faltings
Swiss Federal Institute of Technology - EPFL
Artificial Intelligence Laboratory
vincent.schickel-zuber@epfl.ch, boi.faltings@epfl.ch

Abstract

Various approaches have been proposed to quantify the similarity between concepts in an ontology. We present a novel approach that allows similarities to be asymmetric while still using only information contained in the structure of the ontology. We show through experiments on the WordNet and GeneOntology that the new approach achieves better accuracy than existing techniques.

1 Introduction

With the emergence of the semantic web and the growing number of heterogenous data sources, the benefits of ontologies are becoming widely accepted. The domain of application is widening every day, ranging from word sense disambiguation to search of biological macromolecules such as DNA and proteins.

Initially, ontologies were used in an attempt to define all the concepts within a specific domain and their relationships. An example of a popular ontology is *WordNet* [Miller *et al.*, 1993], which models the lexical knowledge of a native English speaker. Information in WordNet is organized around lexical groupings called synsets and semantic pointers. Informally, a synset represents a set of synonym words, while a semantic pointer models the semantic relationships between two synsets. E-commerce sites such as Amazon.com or Yahoo also use simple taxonomies to classify their products. More recently, biologists and computer scientists have developed an ontology named *GeneOntology* [GO, 2000], which models biological processes, molecular functions and cellular components of genes.

Nowadays, the usage of ontologies goes far beyond domain specification. A very promising direction for ontologies is *semantic search* [Guha, 2003], where ontologies can successfully be used to find documents [Davies, 2002], pictures [Janecek, 2005], and jobs [Bradley *et al.*, 2000]. Semantic search is in fact an information retrieval application, where semantic knowledge captured by an ontology is used to enrich the available vocabulary. In comparison, traditional information retrieval applications use the Vector Space Model (VSM, [Frakes, 1992]) to represent the set of possible items and input query into a common vector space in order to compute the similarity between them. Unfortunately, if no document contains any of the input keywords, the VSM approach fails to find any relevant documents. To overcome this problem, semantic search uses domain ontologies to explore concepts similar to the stated keywords in order to build smarter search engines.

As highlighted by the previously stated examples, evaluating semantic similarity between concepts is a fundamental task. Most authors have focused their research on hierarchical ontologies (HO), which is not surprising as most ontologies are made of *is-a* relationships ($\simeq 82\%$ of all the relations in WordNet 2.0 are *is-a* relationships, while GeneOntology has $\simeq 87\%$). Furthermore, [Maguitman, 2005] has shown that a similarity metric defined on a hierarchical ontology can be generalized to any kind of ontology by using a weighted combination of the *is-a* metrics. Thus, the problem of evaluating the semantic similarity between concepts in any kind of ontology can be simplified to hierarchical ontologies. To this day, there exist two main approaches for estimating similarity between concepts in a hierarchical ontology: the edge based and the node based approaches. Unfortunately, existing approaches fail to achieve high correlation with human ratings, while experiments on the GeneOntology have shown that no technique is best everywhere.

In this paper, we define a novel similarity measure for hierarchical ontologies called *Ontology Structure based Similarity* (OSS). OSS computes the similarity between two concepts a and b in three basic steps. First, we start by inferring the *score* of the concept b from a. From this inferred score, we compute how much has been transferred between these two concepts. Finally, we apply a distance function that transforms the transfer of score into a distance value.

The remainder of the paper is organized as follows. First we review the most popular similarity metrics in Section 2, while our OSS approach is defined in Section 3. Then, in Section 4, we present experimental results of OSS on the WordNet and GeneOntology. Finally, Section 5 concludes this paper.

2 Existing Measures of Similarity

Many techniques have been proposed for evaluating the semantic similarity between two concepts in a HO. They can be classified into two categories: edge based and node based approaches. At the same time, authors have looked at this problem from either a $distance$ or $similarity$ point of view.

These approaches are duals, as the $similarity$ can be defined as $1 - distance$ when values are normalized to [0..1].

The edge based approach is the traditional, most intuitive, and simplest similarity measure. It computes the distance between two concepts based on the number of edges found on the path between them. [Resnik, 1995] introduced a variant of the edge-counting method, converting it from a distance to a similarity metric by subtracting the path length from the maximum possible path length:

$$sim_{EDGE}(a, b) = (2 \times D) - len(a, b) \tag{1}$$

where a and b are concepts in HO, D is the maximum depth of the HO, and $len(a, b)$ is the shortest path between concepts a and b. Another popular variant of the edge based approach is the metric proposed by [Leacock, 1997], which scales the shortest path by twice the maximum depth of the HO.

$$sim_{LEACOCK}(a, b) = -log\left(\frac{len(a, b)}{2 \times D}\right) \tag{2}$$

The node based approach was proposed by [Resnik, 1995] to overcome the drawbacks of the edge-counting approach, which considers the distance uniform on all edges. Resnik defined the similarity between two concepts as the *information content* of the lowest common ancestors, $LCA(a, b)$. The information content (IC) of a concept c is defined as the negative log likelihood of the probability of encountering an instance of the concept, i.e. $IC(c) = -logP(c)$. The intuition behind the use of the negative likelihood is that the more probable a concept is of appearing, then the less information it conveys. Formally, the similarity is defined as follows.

$$sim_{RESNIK}(a, b) = \max_{c \in LCA(a,b)} IC(c) \tag{3}$$

While Resnik defined the similarity based on the shared information, [Lin, 1998] defined the similarity between two concepts as the ratio between the amount of information needed to state the commonality between these two concepts and the information needed to fully describe them.

$$sim_{LIN}(a, b) = \frac{2 \times IC(LCA(a, b))}{IC(a) + IC(b)} \tag{4}$$

Hybrid approaches combine both approaches defined above. [Jiang and Conrath, 1997] proposed a combined model that is derived from the edge based notion by adding the information content as a decision factor. They defined the link strength between two concepts as the difference of information content between them. Following this, Jiang's distance metric is defined as follows:

$$dist_{JIANG}(a, b) = IC(a) + IC(b) - 2 \times IC(LCA(a, b)) \tag{5}$$

3 The OSS Approach

In this section, we define a novel similarity measure for hierarchical ontologies called *Ontology Structure based Similarity* (OSS).

As shown in Figure 1, OSS computes the similarity between two concepts a and b in three basic steps. First, we start by inferring the *score* of the concept b from a, $S(b|a)$. It is based on assigning concepts in the ontology an *a-priori score* (APS), and computing the relations between the scores assigned to different concepts. From this inferred score, we compute how much has been transferred between these two concepts, $T(a, b)$. Finally, we transform the transfer of score into a distance value $D(a, b)$.

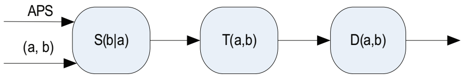


Figure 1: The OSS approach

3.1 Definitions and Assumptions

In this work, an ontology λ is defined as a directed acyclic graph, (DAG), where a node represents a primitive concept, while an edge models the binary specialization relation ($is-a$) between two concepts. Thus, the ontology establishes a hierarchy where each concept can have a set of sub-concepts known as the *descendants*.

Furthermore, a concept represents *instances* (i.e.:a group of objects) with the same *features*, but not all instances of a concept must belong to a sub-concept. Consequently, instances of different sub-concepts are distinguished by differences in certain features. However, these are usually not made explicit in the ontology. Concretely, we see a feature as a restriction on a property or a combination of properties that differentiates a concept from its parent. For example, the subclasses of red and white wines are distinguished by a combination of features that include color and also certain aspects of taste.

We believe that the primary objective of a similarity measure is to simulate a user's behavior and closely correlate with it. In particular, a quantitative measure of similarity should express the ratio of numerical scores that may be assigned to each concept. The score could reflect how much an item is preferred, or how friendly it is to the environment. For this purpose, we assume that the *score* S of a concept is a real-valued function normalized to $[0..1]$ that satisfies the following assumptions:

- A1: the score depends on features of the concept.
- A2: each feature contributes independently to the score.
- A3: unknown and disliked features make no contribution to the score, i.e. the score of a concept is a lower bound on the possible score of its instances.

Assumption A1 is very intuitive and reflects the fact that a concept is modeled by a set of features. Thus, the score will only be influenced by the features making up the concept. It is also the basis of multi-attribute decision theory (MAUT - [Keeney and Raiffa, 1993]), where the utility of an item depends on the preference value of the attributes making that item. Thus, all instances of the same concept will have the same score as they share the same features.

The second assumption eliminates the inter-dependence between the features and allows the score to be modeled as the sum of the scores assigned to each feature. In MAUT, an even stronger assumption (the mutual preferential independence) is used to build an additive value function for an item. Independence is a strong assumption, but it is still more accurate that the assumption that all the features correlate positively to the score.

The third assumption may appear counterintuitive, but it reflects the observation that users are risk averse. For example, if the score models the price that a user is willing to pay for an item, it is rational for users to adopt this pessimistic view, since one would not normally be willing to pay for features that have not been explicitly provided. Thus, the score attached to a concept can be seen as a lower bound on the score that items belonging to that concept might have.

3.2 Computing an A-priori Score

An ontology is usually designed in such a way that its topology and structure reflects the information contained within and between the concepts. A major ingredient of OSS is the computation of the *a-priori score* of a concept c, $APS(c)$, which captures this information. The APS models the expected score of each concept for an average user, but without using any user information. It is not used as a prediction of actual scores, but only to estimate constants (α and β) that determine how actual users scores propagate through the ontology.

As we have no information about the user, we assume that all concepts have a score that is uniformly distributed between 0 and 1. This is often a reasonable assumption as each concept exists to satisfy the desire of some group of people. Thus, the probability that the score of a concept c is superior to the threshold x, ($S(c) > x$), is equal to $1 - x$. However, this probability ignores the fact that concepts can have descendants. Furthermore, our model is by definition pessimistic (A3), which means that the score should be a lower bound of the score of its instances and the score of its descendants. Therefore, the probability that the score of any concept c is superior to a threshold x is equal to $(1-x)^{n+1}$, where n is the number of descendants of c. Note that we count all descendants, not just the leaves, to account for the fact that each concept has instances that do not belong to any sub-concept.

Following this, the probability distribution of the score for a concept c is $P(S(c) \leq x) = 1 - (1-x)^{n+1}$, while the expected score can be obtained by integration of the density function of c. Formally, $E(S)$ is defined as follows.

$$\begin{aligned} E(S) &= (n+1)\int_0^1 x(1-x)^n dx \\ &= (n+1)\left[\underbrace{\left|-\frac{x(1-x)^{n+1}}{n+1}\right|_0^1}_{0} + \underbrace{\int_0^1 \frac{(1-x)^{n+1}}{n+1}}_{1/((n+1)(n+2))}\right] \qquad (6) \\ &= \frac{1}{n+2} \end{aligned}$$

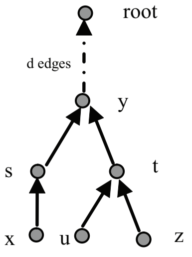
Equation 6 tells us that the expected score of a concept c will be inversely proportional to the number of its descendants + 2. Following this, we define the a-priori score of a concept c with n descendants as:

$$APS(c) = \frac{1}{n+2} \qquad (7)$$

The a-priori score defined in Equation 7 implies that the leaves of the ontology will have an APS equal to $1/2$, which is equal to the mean of a uniform distribution between 0 and 1. Conversely, the lowest values will be found at the root. This means that when we travel up the ontology, the concept becomes more generalized, and therefore the APS decreases. Another important aspect of this APS is the fact that the difference in score between concepts decreases when we travel up the ontology, due to the increasing number of descendants.

Resnik also uses the topology to compute the *information content* of a concept. The APS share some similarities with information content. For example, the difference in both APS and IC decreases when we travel up the ontology. However, some profound differences exist. First, the APS is a bottom-up approach that considers the differences between the concepts, while Resnik's is a top-down approach that considers the commonalities between two concepts. Second, we use the $1/x$ function to compute our score, while Resnik uses the logarithm to base 2. In the validation section, we show that this brings better results than the information content approach.

To illustrate the computation of the a-priori score, consider the simple ontology λ shown in Figure 2(a). First, the number of descendants of each concept n_c is computed. Then, we apply equation (7) to compute the APS of each concept in λ.



(a)

Concepts	n_c	APS
x	0	1/2
u	0	1/2
z	0	1/2
s	1	1/3
t	2	1/4
y	5	1/7
root	5+d	1/(7+d)

(b)

Figure 2: (a) a simple ontology λ and its APSs (b)

3.3 Inferring the Score

Re-consider the ontology λ shown in Figure 2(a), and imagine a situation where we want to compute the similarity between the concepts x and z. To propagate the score between these concepts, a link between them must be found. Thus, the first task in the propagation is to identify the chain that contains both concepts. To minimize the propagation, we construct the chain through the lowest common ancestor, LCA.

In a tree graph, a lowest common ancestor is defined as the closest upward reachable node shared by x and z [Knappe *et al.*, 2003]. Note that in a DAG, there can be several LCA nodes; in fact, the number of LCA nodes can grow exponentially with the size of the graph. Fortunately, this number tends to be small in reality, as most concepts have only a few parents. For example, a concept in the WordNet ontology has on average 1.03 parents.

We use the following heuristic method to select which LCA to use to propagate the score. For each possible LCA node y, we compute the following values:

- its depth $d(y)$, given as the distance of the longest path between the *root* and y, and
- its reinforcement $r(y)$, given as the number of different paths leading to y from x and z.

We pick the LCA as the node with the highest value of $r(y) * 2^{d(y)}$. The idea behind this heuristic is to consider the fact that a concept found higher in the ontology (low $d(y)$) can still be more meaningful to the user if it has many paths leading to it (high $r(y)$).

Upward Inference $\nearrow$

The first situation that arises is when there is a path going from concept x to its k^{th} parent y. From the construction of the ontology, both concepts have d features in common but the concept x has an extra k features that differentiate it from its ancestor. By definition of the model, we know that the score of a concept depends on the features defining that concept (A1). Informally, it means that the score of y can be estimated knowing the score of x, $S(y|x)$, by looking at the ratio of features they have in common. We define $S(y|x)$ as:

$$S(y|x) = \alpha(x,y)S(x) \tag{8}$$

where $\alpha(x,y)$ is a coefficient of generalization. For every pair of concept x and ancestor y in the ontology, we estimate α using their ratio of a-priori score.

$$\hat{\alpha}(x,y) = APS(y)/APS(x) \tag{9}$$

Downward Inference $\searrow$

Inversely, we have the case where z is the l^{th}descendant of y. Following Equation 8, it is very tempting to assume that $S(z|y) = \beta S(y)$, where β is a coefficient of specialization that contains the ratio of features in common. However, this reasoning is not compatible with our second assumption – features contribute to the score independently. To understand this assumption, imagine that the score of the object is equal to the maximum price a user is willing to pay. Consider two concepts a and b, where a has one more feature than b. Now consider two users A and B such that A values b more than B does. This does not automatically mean that A will also attach a higher value to the extra feature that distinguishes a from b. Notice also that when we were traveling upwards, we were considering super concepts. This means that we were removing known features whose contribution to the score is likely to be proportional to it. However, when traveling downwards, we are adding new (unknown) features to the concept. Therefore, we need to consider the score of each new feature independently. Formally, it means that $S(z|y)$ should be defined as follows.

$$S(z|y) = S(y) + \beta(z,y) \tag{10}$$

where β is a coefficient of specialization that we estimate using the a-priori score:

$$\hat{\beta}(z,y) = APS(z) - APS(y) \tag{11}$$

3.4 Transfer of Score

Our intuition is that the *distance* between two concepts a and b is correlated to the amount of score being transferred between them.

Formally, a distance measure is a real valued function that we would like to satisfy the following axioms:

- *identity*: $D(a,b) = 0 \Leftrightarrow a = b$
- *normalization*: $0 \leq D(a,b) \leq 1$
- *triangle inequality*: $D(a,c) \leq D(a,b) + D(b,c)$

It is otherwise customary to also include a symmetry axiom; however, one innovation of our proposal is that distance can be asymmetric so we do not include it.

Similarity and distance between two concepts must furthermore be independent of any particular amount of score that is being assigned to the concept. In particular, it must be applicable to any score in the admissible range [0..1]. Thus, distance can be related to the transfer of score only in a multiplicative, but not in an additive way. Following this, we define the *transfer of score* from a concept a to b, $T(a,b)$, as the amount of score being transferred from a to b, i.e.:

$$S(b|a) = S(a) \times T(a,b) \Rightarrow T(a,b) = \frac{S(b|a)}{S(a)} \tag{12}$$

Following our example, and using Equation (8) and (10), $T(a,b)$ can be decomposed as follows.

$$\begin{aligned} T(x,y) &= \alpha(x,y) \qquad \nearrow \\ T(y,z) &= 1 + \tfrac{\beta(z,y)}{S(x)} \qquad \searrow \end{aligned} \tag{13}$$

In our context, $S(x)$ is unknown, which renders the above computation for downwards transfer impossible. However, under the assumption that the score of any concept is uniformly distributed between 0 and 1 (Section 3.2), the expected score of a concepts is in fact equal to $^1/_2$. Thus, we approximate $T(y,z)$ as $1 + 2\beta(z,y)$.

When judging the distance of two concepts a and b, the transfer of score from a to b should not attempt to predict the expected value of the score assigned to b, but a lower bound on it in order to model the fact that unknown features do not contribute. For upward propagation, the factor α derived from the a-priori scores reflects this correctly, as the APS of an ancestor is constructed as a lower bound on scores of its descendants.

On the other hand, for downward propagation, the term β derived from the a-priori score reflects that translation from the lower bound to the expected value at the descendants. Thus, in order to produce a lower bound, the relation has to be inverted.

Consequently, the transfer $T(a,b)$ becomes as follows.

$$\begin{aligned} \hat{T}(x,y) &= \hat{\alpha}(x,y) \qquad \nearrow \\ \hat{T}(y,z) &= \tfrac{1}{1+2\hat{\beta}(z,y)} \qquad \searrow \end{aligned} \tag{14}$$

3.5 Similarity Metric

We define the *Ontology Structure based Similarity* similarity $sim_{OSS}(a,b)$ of two concepts a and b as $1 - D(a,b)$, where $D(a,b)$ is a distance measure between the concepts.

To guarantee that $D(a,b) \geq 0$, and in order to satisfy the identity relation, we transform the transfer of score into a distance by taking its negative logarithm: $D(x,z) = -\log T(x,z)$. However, this distance would not be normalized to fall in the interval [0..1]. We thus normalize to obtain:

$$D(a,b) = -\frac{\log(T(a,b))}{\max D} \tag{15}$$

where $\max D$ is longest distance between any two concepts in the ontology. Thus, the distance measure satisfies the normalization. Such a logarithmic measure has also been used elsewhere for distance and similarity measure [Jiang and Conrath, 1997][Lin, 1998][Resnik, 1995].

To verify that the distance measure satisfies the triangle inequality, we consider the transfer from concept x to z (Figure 2(a)), and the additional concepts s and t. Assume that the ontology is tree-structured, so that there is a unique path from x to z with LCA y. Note that $D(x,z) = D(x,y) + D(y,z)$.

Assume first that s is a node on the path from x to y. Then if s is part of the upward path from x to y, equation (9) implies that $T(x,s) = APS(s)/APS(x)$ and $T(s,y) = APS(y)/APS(s)$. Furthermore, and because $T(x,y) = APS(y)/APS(x)$, we can show that

$$\begin{aligned} -\log(T(x,y)) &= -\log(T(x,s) \times T(s,y)) \\ &= -\log(T(x,s)) - \log(T(s,y)) \end{aligned} \tag{16}$$

As a consequence, $D(x,y) = D(x,s) + D(s,y)$ and thus the triangle inequality holds as $D(x,z) = D(x,s) + D(s,y) + D(y,z)$. If t is part of the downward path from y to z, then we get that $T(y,t) = 1/(1+2\beta(t,y))$ and $T(t,z) = 1/(1+2\beta(z,t))$. By definition, $T(y,z)$ is equal to $1/(1+2\beta(z,y))$, and Equation (11) implies that $\beta(z,y) = \beta(z,t) + \beta(t,y)$. Thus, we get the following equation:

$$
\begin{aligned}
-log(T(y,z)) &= -\log\left(\frac{1}{1+2\beta(z,y)}\right)\\
&\leq -\log\left(\frac{1}{1+2\beta(z,y)+4\beta(z,t)\beta(t,y)}\right) \quad (17)\\
&= -\log\left(\frac{1}{1+2\beta(z,t)} \times \frac{1}{1+2\beta(t,y)}\right)\\
&= -\log(T(y,t)) - \log(T(t,z))
\end{aligned}
$$

which shows that the triangle inequality also holds when t is on the downwards path connecting y and z.

Now consider the case where s is not on the path from x to z. Due to the tree structure, the paths from x to s and from s to z will both contain a common part between a node s and a node d that is on the path from x to z. Since all transfers are ≤ 1, this part can only increase the combined distance, so the triangle inequality will still hold.

Using Equations (14) and (15), we thus defined the distance between any two concepts in the ontology x and z, given the LCA y as follows.

$$D(x,z) = \frac{\log(1+2\hat{\beta}(z,y)) - log(\hat{\alpha}(x,y))}{\max D} \quad (18)$$

Table 1 illustrates the distance computation between the concepts x and z of the example in Figure 2(a).

Concepts	Direction	Transfer	Distance	$D(x,y)$
x $\curvearrowright$ z	x $\nearrow$ y	$\hat{\alpha} = \frac{1}{7}/\frac{1}{2}$	$-\log(\frac{2}{7})$	$\simeq \frac{2.58}{maxD}$
	y $\searrow$ z	$1+2\hat{\beta} = \frac{24}{14}$	$\log\left(\frac{24}{14}\right)$	

Table 1: Distance between concepts x and z in ontology λ

4 Experiments

4.1 Experiment I - WordNet

When Resnik introduced the node-based approach, he also established an evaluation procedure that has become widely used ever since. He evaluated his similarity metric by computing the similarity of word pairs using the WordNet ontology, and then considered how well it correlated with real human ratings of the same pairs. These word pairs were selected in such a way that they covered high, intermediate, and low levels of similarity.

WordNet is the most widely used and one of the biggest ontologies in the world ($\simeq$80000 concepts), which makes experiments credible. As many authors in the field do, we reproduced Resnik's experiment with the WordNet 2.0 on the original 30 word pairs. The correlations between various metrics and human ratings are displayed in Table 2.

Edge	Leacock	Resnik	Lin	Jiang	OSS
0.603	0.823	0.793	0.823	0.859	**0.911**

Table 2: Correlation with various similarity metrics

Our approach using the a-priori score achieves over 91% correlation with real user ratings, and clearly demonstrates significant benefit over earlier approaches (t-obs $\simeq$ 3.28 and p-value $<$ 0.02).

As expected, the hybrid approach performed better than existing techniques, but the improvement over the information based approach was not statistically significant (t-obs = 1.46 and p-value $\simeq$ 0.08). The edge based approach is the worst performing metric as it supposes that the edges represent uniform distances, which is obviously not true in WordNet.

$\nearrow$	α	α	$1+2\beta$	$1+2\beta$
$\searrow$	$1+2\beta$	α	α	$1+2\beta$
Corr.	0.911	0.882	0.693	0.785

Table 3: Different combinations of the coefficients in OSS

We tried different combinations of the coefficients α and β in order to test the upward and downward propagation. As expected, Table 3 shows that the best correlation is obtained when using α going up and $1+2\beta$ going down. As mentioned earlier, these coefficients renders the metric asymmetric. In fact, experiments showed that the upwards distance is up to 15 times greater the the downwards distance when concepts are very dissimilar.

In section 3.4, we estimated the downward transfer between two concepts by using the expected score of a concept, i.e.: $S(x) = {}^1\!/_2$. To verify this assumption, we tested our metric with various scores ranging from the minimum APS in the ontology to 1.

$1+\beta$	$1+\frac{4}{3}\beta$	$1+2\beta$	$1+4\beta$	$1+\frac{1}{minAPS}\beta$
0.910	0.910	0.911	0.910	0.814

Table 4: Different value for S(x) in OSS

Table (4) shows that the optimum value does in fact occur when $S(x) = {}^1\!/_2$, and any value around it will not greatly influence the correlation. However, big underestimations of the initial score tend to influence the correlation by over 10% as it will overestimate the coefficient β.

4.2 Experiment II - GeneOntology

To show the generality of the results, we performed another experiment on a much bigger scale using the GeneOntology (GO). GO was chosen over others as it is one of the most important ontology within the bioinfomatics community, and with over 20000 concepts, it is also one of the biggest.

As the name suggested, GeneOntology is an ontology describing gene products. Formally, the ontology is a DAG, where a concept represents a gene, and where an edge models *is-a* or *part-of* relationships. By definition, a gene product might be associated with or located in one or more cellular components; it is active in one or more biological processes, during which it performs one or more molecular functions. Thus, the DAG is further decomposed into three orthogonal sub-ontologies: *molecular function* (MF), *biological process* (BP), and *cellular component* (CC).

As for most real life applications, there is no human data of similarity over which we could benchmark our metric. Instead, [Lord *et al.*, 2003] proposed to use the Basic Local Alignment Search Tool (BLAST - [Altschul, 1990]) as it shows some correlations with concept similarity. BLAST compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. In another word, BLAST finds regions of local similarity between sequences.

Formally, the experiment was conducted a follows. First, we downloaded the February 2006 releases of the SWISS-PROT protein database[1] (SPD), GO[2], and BLAST[3]. Then, we went through all the concepts in GO, and selected each concepts that was also present in SPD. To reduce the noise and errors, we also removed all the proteins that were not annotated by a traceable authors (evidence code = TAS). For all remaining concepts, we performed a BLAST search[4] in order to get a list of similar proteins. From this list, three proteins were randomly selected; respectively one at the beginning, the middle and one at the end. The idea behind this selection is to cover high, intermediate, and low levels of concept similarity. During a search, BLAST associates a $score$ to each result that measures the similarity with the input protein. Therefore, it is this score (after normalization) that has been used as benchmark measure.

After the BLAST searches, we measured the similarity between the concepts representing the input proteins and the result ones using all of the metrics mentioned in this paper. Finally, we measured the deviation between the normalized BLAST score and the similarity values of all the concepts using the *mean absolute error* measure (MAE - [Herlocker *et al.*, 2004]). For each of GO's sub-ontologies, Table 5 shows the deviation values (MAE) for all the similarity metrics.

	Edge	Leacok	Resnik	Lin	Jiang	OSS
MF	0.450	0.234	0.224	0.223	*0.200*	**0.185**
BP	0.392	0.275	0.314	0.312	*0.269*	**0.259**
CC	0.351	0.303	*0.286*	0.292	0.343	**0.260**

Table 5: MAE of various similarity metrics on GO

The results are very interesting in two points. First, it shows that none of the existing techniques dominates another one. For example, Jiang's metric has a lower deviation on the MF ontology than Resnik's metric, but it is not true for the CC ontology. These results can be explained by the fact that the topology of the sub-ontologies differ widely. For example, BP has 10796 concepts and 85.3% of *is-a* relations, MF has 7923 concepts and 99.9% of *is-a* relations, while CC has only 1181 concepts and 59.8% of *is-a* relations.

Finally, we can see that the OSS approach has the lowest deviation, whatever the sub-ontology. This suggests that our approach is more robust than existing techniques, and that it is also more accurate.

5 Conclusion

This paper makes two major contributions. First, we showed that a-priori scores can be used to exploit the implicit knowledge captured by the ontology. Second, we introduced a new technique called Ontology Structure Similarity to derive a similarity metric based on these a-priori scores. The similarity metric exploits the implicit knowledge of the person who wrote the ontology and gave it a certain structure. A major novelty is that similarities and distances are asymmetric. Experimental evaluation has shown that OSS outperforms existing techniques on WordNet and GeneOntology.

References

[Altschul, 1990] Gish-W. Miller W. Myers E.W. Lipman D.J. Altschul, S.F. Basic local alignment search tool. In *Jour. of molecular biology*, volume 215, pages 403–410, 1990.

[Bradley *et al.*, 2000] K. Bradley, R. Rafter, and B. Smyth. *Case-Based User Profiling for Content Personalization (AH2000)*, pages 133–143. Springer-Verlag, 2000.

[Davies, 2002] Weeks-R. Krohn U. Davies, J. Quizrdf: Seach technology of the semantic web. In *Proceedings of the Int. Workshop on Real Word RDF WWW'02*, 2002.

[Frakes, 1992] Baeza-Yates R. Frakes, W. *Information Retrieval: Data Structures & Algorithms*. 1992.

[GO, 2000] The Gene Ontology Consortium GO. Gene ontology: tool for the unification of biology. In Nature America, editor, *Nature Genetic*, volume 25, pages 25–29, 2000.

[Guha, 2003] McCool R. Miller E. Guha, R. Semantic search. In *12th Int. Conf. on WWW*, pages 700 – 709, 2003.

[Herlocker *et al.*, 2004] J. L. Herlocker, J. A. Konstan, L. G. Terven, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. In *ACM Transactions on Information Systems*, volume 22, pages 5 – 53, 2004.

[Janecek, 2005] Schickel-Zuber V. Pu P. Janecek, P. *Concept Expansion Using Semantic Fisheye Views*, volume LNCS 3815, pages 273–282. Springer-Verlag, 2005.

[Jiang and Conrath, 1997] J. Jiang and D.W Conrath. Semantic Similarity based on corpus and lexical taxonomy. In *Proc. of 10th Int. Conf. COLING*, 1997.

[Keeney and Raiffa, 1993] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preference and Value Tradeoffs*. Cambridge University Press., 1993.

[Knappe *et al.*, 2003] R. Knappe, H. Bulskov, and T. Andreasen. Similarity graphs. In *Proceedings of ISMIS'03*, volume LNAI 2871, pages 668 – 672. Springer, 2003.

[Leacock, 1997] Chodorow-M. Leacock, C. Combining local context and WordNet similarity for word sense identification. In *Fellbaum*, pages 265 – 283, 1997.

[Lin, 1998] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th Int. Conf. on Machine Learning*, 1998.

[Lord *et al.*, 2003] P.W. Lord, R.D. Stevens, A. Brass, and C.A Goble. Semantic Similarity Measures as Tools For Exploring The Gene Ontology. In *Pacific Symposium on Biocomputing*, volume 8, pages 601 – 612, 2003.

[Maguitman, 2005] Menczer-F. Roinestad H. Vespignani A. Maguitman, A. Algorithmic Detection of Semantic Similarity. In *Proceedings of the Int. World Wide Web*, 2005.

[Miller *et al.*, 1993] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: An On-line Lexical Database. Technical report, Cognitive Science Laboratory, Princeton University, 1993.

[Resnik, 1995] P. Resnik. Using information content to evaluate semantic similarity. In *Proceedings of the IJCAI05*, pages 448 – 453, 1995.

---

[1] ftp://ftp.ebi.ac.uk/pub/databases/

[2] http://www.geneontology.org/GO.downloads.shtml#ont

[3] http://ncbi.nih.gov/BLAST/download.shtml

[4] blastall -p blastp -d swissprot -i in.txt -o out.txt -e 1000 -v 1000