Measuring Semantic Similarity using a Multi-Tree Model

Behnam Hajian and **Tony White**
School of Computer Science Carleton University, Ottawa, Canada
{bhajian,arpwhite}@scs.carleton.ca

Abstract

Recommender systems and search engines are examples of systems that have used techniques such as Pearson's product-momentum correlation coefficient or Cosine similarity for measuring semantic similarity between two entities. These methods relinquish semantic relations between pairs of features in the vector representation of an entity. This paper describes a new technique for calculating semantic similarity between two entities. The proposed method is based upon structured knowledge extracted from an ontology or a taxonomy. A multi-tree concept is defined and a technique described that uses a multi-tree similarity algorithm to measure similarity of two multi-trees constructed from taxonomic relations among entities in an ontology. Unlike conventional linear methods for calculating similarity based on commonality of attributes of two entities, this method is a non-linear technique for measuring similarity based on hierarchical relations which exist between attributes of entities in an ontology. The utility of the proposed model is evaluated by using Wikipedia as a collaborative source of knowledge.

1 Introduction

Similarity refers to psychological nearness between two concepts. Similarity has roots in psychology, social sciences, mathematics, physics and computer science [Larkey and Markman, 2005]. In social psychology, similarity points to how closely attitudes, values, interests and personality match between people which can lead to interpersonal attraction. This can be explained by the fact that similar people tend to place themselves in similar settings and this consequently decreases potential conflicts between them. Furthermore, finding a person with similar tastes helps to validate values or views held in common. With a mental representation, the similarity between two concepts is defined as a function of the distance between two concepts represented as different points in the mental space [Tversky and Shafir, 2004].

Semantic similarity is used to refer to the nearness of two documents or two terms based on likeness of their meaning or their semantic contents [Tversky and Shafir, 2004]. Conventionally, statistical means (e.g., a vector space model) can estimate the distance between two entities by comparing features representing entities [Salton *et al.*, 1975]. For example, in order to compare two documents, the frequency of co-occurrence of words in the text corpus represents the similarity between the two documents. Semantic relatedness is a broader term than semantic similarity, with the former including other concepts such as antonymy and meronymy. However, in certain literature these two terms are used interchangeably .

We can compare the similarity of two concepts by measuring the commonality of their features. Since each concept is represented by the features describing its properties, a similarity comparison involves comparing the feature lists representing that concept. Simply put, concepts which are near to each other are more similar than points which are conceptually distant. There are several mathematical techniques for estimating this distance, such as latent semantic analysis (LSA) [Landauer *et al.*, 1998]. Measuring similarity among entities has applications in many areas such as: recommendation systems, e-commerce, search engines, biomedical informatics and in natural language processing tasks such as word sense disambiguation.

For instance, in user-based collaborative filtering, the system tries to find people with similar tastes and recommend items highly ranked by the people which might be interesting to their peers. Finding people with similar tastes involves processing of their historical transactions (i.e., items viewed and ranked by them in their previous transactions) and calculating similarity between them using one of the methods described above. On the other hand, in content-based recommender systems and search engines, the system finds items which are more similar to show to a user based on his/her query and the similarity of the items (i.e., products). This category of system calculates similarity between products based on the commonality of the features of different products.

In information retrieval (IR) and search engines, words are considered as features in a document or a query. In IR systems, it is conventional to represent a document by a bag-of-words (BOW). A Vector Space Model (VSM) is generally used to estimate the similarity between two documents in classification/clustering tasks or to estimate similarity between a query and documents in keyword-based search engines.

1.1 Contribution, Motivations and Paper Structure

In the Vector Space Model (VSM), a document or a query is represented as a vector of identifiers such as index terms. However, in many cases, conventional methods such as Dices coefficient, Pearson's correlation coefficient, Jaccards index or cosine similarity, which use VSM to represent a document, do not perform well. This is due to a document being represented in a linear form (i.e., a vector of features) in which semantic relations among features are ignored. Examples of such problems are: ignoring polysemy (terms having different sense with a same spelling such as apple as a fruit and apple as a company) and synonymy (terms with different spelling having a same sense such as big and large). An example of the latter problem can be found in recommender systems which find people with similar tastes according to their previous transactions. An example of this problem is demonstrated in the following example in which similarity of tastes for two people are estimated based on their previous transactions:

- T1 (Clothes, Boxspring, Mp3Player, Mattress, LCD TV)
- T2 (Dress, Bed, Mattress, iPod Touch, LED TV)

Using the VSM-based method for computing similarity between the above transactions, these transactions are no longer similar at all. However, intuitively we have a feeling that LED TV and LCD TV are related to each other since both are subclasses of TV. This observation is also true when comparing iPod Touch and Mp3 Player and for the relationship between Clothes and Dress.

This paper proposes replacing the VSM with a non-linear representation for an entity. The proposed representation models an entity by its features in a hierarchical format using an ontology called a semantic multi-tree. Multi-tree similarity considers semantic relations among the features of entities in a hierarchical structure using the ontology classification. This method enhances conventional information retrieval techniques by computing similarity regarding commonality of not only the features but also commonality of semantic relations among features and their parents.

The rest of the paper is organized as follows: The next section provides background information on the VSM including analysis of its limitations as well as related work. In Section 3, we define our model for representing entities called the semantic multi-tree. Section 4 concentrates on the definition of the proposed method for measuring similarity by use of a semantic multi-tree model. In the following section, the technique is validated against human judgment in WordSimilarity-353 and Rubenstein and Goodenough using Wikipedia categories as a taxonomy. A discussion follows, with conclusions and future work provided in the final section.

2 Background and Related Work

The Vector Space Model is defined as an algebraic model in which a document or an entity is represented as a vector of its features; for example, a document which is represented as a vector of index terms [Salton and McGill, 1983]: $d_j = (w_{1j}, ..., w_{nj})$. In these vectors, w_{ij} represents the number of occurrences of the i^{th} term in the j^{th} document. There are two model representation schemes. The superior scheme for representation of vectors in this model is term frequency-inverse document frequency (tf-idf). In the other scheme, w_{ij} is a binary representation of the occurrence of a corresponding term. In order to retrieve a document among a collection of documents, we have to calculate the similarity between our query and all of the documents in the collection and choose the most relevant documents among them. A frequently used method for estimating the similarity is calculating the cosine of the angle between the vectors representing query and a document. The higher the cosine of the angle between two vectors the more similar the vectors and, therefore, the more similar the entities represented by the vectors.

$$cos\theta = sim(d_i, q) = \frac{d_i.q}{|d_i||q|} \tag{1}$$

Another method for calculating similarity is Pearson product-moment correlation coefficient (PMCC). This method calculates the correlation (linear dependence) between two vectors. The PMMC of two vectors is defined as:

$$sim(d_i, q) = \frac{cov(d_i.q)}{\sigma d_i \times \sigma q} \tag{2}$$

Calculation of similarity is straightforward with these methods but a disadvantage is that neither of them considers semantic relations among features.

A conventional method for computing semantic relatedness is the corpus-based technique that relies on the tendency for related words to appear in similar texts called Latent Semantic Analysis (LSA). Unfortunately, LSA is only able to provide accurate results when the corpus is very large.

In recent years, several techniques have been developed – such as the work proposed by Ted Pedersen et. al – for estimating similarity by measuring semantic distance between two words in WordNet [Pedersen *et al.*, 2005]. A limitation of using lexical databases such as WordNet or similar resources is that they have been created by one or a group of linguists rather than experts in different subjects. Furthermore, WordNet is rarely revised when compared to collaborative knowledge sources such as Wikipedia. As a result, WordNet does not include some special proper nouns in different areas of expertise (e.g., Obama, Skyneedle). Recently, Wikipedia has compensated for this lack of knowledge by providing a mechanism for collaboratively creating knowledge. Wikipedia includes a wide range of articles about almost every entity in the world by using human expertise in different areas. In addition, as of May 2004, Wikipedia articles have been categorized by providing a taxonomy; namely, categories. This facility provides hierarchical categorization with multiple parents for a node by means of a multi-tree structure. This observation motivates us to use Wikipedia as a resource of knowledge in this paper.

There are several approaches for measuring semantic relatedness using resources such as WordNet or Wikipedia categories as a graph or network by considering the number or length of paths between concepts. In the WikiRelate project,

Ponzetto and Strube used three measures for computing semantic relatedness: First, a path-based measure using the length of the path between two concepts; second, an information content-based measure and third, the overlap-based measure which applies the Lesk algorithm that defines the relatedness between two words as a function of the overlap between two contexts defining the corresponding words. In WikiRelate [Strube and Ponzetto, 2006], a pair of Wikipedia pages is first retrieved, then categories they refer to are extracted and finally, the relatedness between two concepts is computed regarding the paths found between two concepts in the Wikipedia categories. In the last step, Ponzetto and Strube calculate relatedness by selecting the shortest path and the paths which maximize the information content-based measure.

In contrast with statistical methods for computing relatedness such as LSA, Gabrilovich and Markovitch proposed Explicit Semantic Analysis (ESA) using meaning in natural concepts derived from Wikipedia [Gabrilovich and Markovitch, 2007]. In this work, they used Wikipedia articles for augmenting the text representation and constructing a weighted list of concepts. They finally used tf-idf and conventional machine learning methods to calculate relatedness between the weighted vectors constructed in the previous steps.

Milne and Witten used cross referencing in the Wikipedia link database in order to obtain semantic relatedness between two concepts called Wikipedia Link-based Measure (WLM) [Witten and Milne, 2008]. In WLM, they used tf-idf using link counts weighted by the probability of occurrence of a term in an article. Almost all of the previous research has used tf-idf and VSM to calculate the relatedness between two sets of features.

3 The Semantic Multi-Tree Model

Semantic tree is a term with several different meanings in computer science. The term is regularly used as an alternative term for a semantic tableaux which is a very well known method in logic (i.e., a resolution method for mechanized reasoning) [Annates, 2005]. Semantic tree in this paper is interpreted as the taxonomy of entities in an ontology.

Taxonomy in the literature is defined as the practice and science of classification. Almost all objects, places, concepts, events, properties and relations can be classified into a taxonomic hierarchy. In the ontological literature, taxonomy refers to a set of concepts with *is-a* (i.e., SubClassOf/InstanceOf) relations between them. Therefore, we consider taxonomy as a narrower concept than ontology since ontology includes broader relations such as *part-of*, *has-a*, *rules*, *axioms* and *events* as well as *classes*, *hierarchical relations* and *attributes*.

Definition 1: A taxonomy, $\mathcal{O}$, is defined as a set of classes, and *is-a* relations between them, $\mathcal{O} = (\mathcal{C}_\mathcal{O}, \mathcal{R}_\mathcal{O})$. In formal logic, the *is-a* relation is defined as a *subclass/instance-of* relation in an ontology:

Subclass/Instance-of: $\forall x : c_i(x) \rightarrow c_j(x)$. Such that $\forall c_i, c_j \in \mathcal{C}_\mathcal{O}, \textit{is-a}(c_i, c_j) \in \mathcal{R}_\mathcal{O}$.

Definition 2: A Multi-Tree is defined as a tree data structure in which each node may have more than one parent. A multi-tree is often used to describe a partially ordered set. In this paper, a taxonomy of concepts is modelled by a multi-tree structure in which each concept may refer to multiple super-concepts. It should be noted that in taxonomies such as Wikipedia Categories and WordNet cycles do exist; however, we have avoided capturing them by breaking edges creating directed cycles and capturing the rest of the graph by using a multi-tree structure. It should also be noted that the definition used here is more general in that the algorithms used allow for the existence of multiple paths between a leaf and root node; i.e., the diamond-free poset requirement is relaxed.

Formally, in this paper, a multi-tree is a directed acyclic graph, $T = (V, E, C, L, M, W, P)$, with hierarchical categorization of its nodes in different levels such that:

- *V* is a set of vertices (nodes), $V = \{v_1, ..., v_n\}$. Each vertex corresponds to a concept in the taxonomy.
- *E* is a set of edges, $E = \{e_1, ..., e_n\}$, (in which $e = \langle v_i, v_j \rangle$ is an ordered set representing an edge from node v_i to node v_j. Each edge represents an *is-a* relation between two concepts c_i, c_j which means (c_i *is-a* c_j). The direction in this digraph is always from a concept (subclass) to its parent (super-class).
- *C* is a set of terms representing concepts which are used as nodes labels.
- *L* is a function mapping V to $\mathbb{R}$ $L : V \rightarrow \mathbb{R}$ assigning a real number to each node. This function is recursively defined as being 1 plus the average value of L for the children of the node. Initially, this function assigns 0 to the leaf nodes.
- *M* is a bijective mapping function mapping V to C ($M : V \rightarrow C$) assigning a label (representing a concept) to a node.
- *W* is a function mapping V to $\mathbb{R}$ ($W : V \rightarrow \mathbb{R}$) which assigns a real number as a weight to each node. This weight is utilized to calculate the similarity between two entities, which will be discussed in the next section.
- *P* is a function mapping E to $\mathbb{R}$ ($P : E \rightarrow \mathbb{R}$) which assigns a real number to each edge as a propagation ratio of each edge. In this paper P was set to 1.

The following functions, properties and operators are defined for a Multi-Tree:

- *leaf(v)* is a function mapping V to $\{true, false\}$ that returns a Boolean value indicating whether a node is a leaf node or not. A leaf node in Multi-Tree does not have any children. A multi-tree may have several leaves.
- *root(v)* is a function mapping V to $\{true, false\}$ that returns a Boolean value indicating whether a node is a root node or not. A Multi-Tree node is a root if it does not have any parents. A multi-tree has only one root node.
- *children(v)* is a function mapping V to P(V) (the power set of V) that returns the set of all the direct children of the node.
- *parents(v)* is a function mapping V to P(V) (the power set of V) that returns the set of all the direct parents of the node.

- $\beta_v = |children(v)|$ is defined as the cardinality of the child set of node v. (count of children of the node v)
- $\gamma_v = |parents(v)|$ is defined as the cardinality of the parent set of node v. (count of parents of the node v)
- The combination operator with the symbol $\uplus$ is defined between two multi-trees T_1, T_2 and returns a multi-tree T_u containing all the vertices and edges that exist in both T_1 and T_2. In other words, this operator returns the combination of two multi-trees. $T_u = T_1 \uplus T_2 \Rightarrow$

$$T_u = \begin{cases} E_u = E_1 \cup E_2 \\ V_u = V_1 \cup V_2 \\ C_u = C_1 \cup C_2 \end{cases} \begin{cases} L^{T_u} \\ M^{T_u} \\ P^{T_u} \\ W^{T_u} \end{cases} \quad (3)$$

- The weights of the vertices in the tree T_u are calculated by a recursive function $W^{T_u} : V \times \mathbb{R} \rightarrow \mathbb{R}$ as defined in equation 4. In the proposed algorithm, weight is propagated from the leaves to the root of the multi-tree combined from two multi-trees. In this equation, α is a damping factor (degradation ratio). The damping factor causes the nodes at lower levels of a multi-tree (i.e., nodes near to the leaves) to contribute more to the weight than nodes in higher levels.

$$W^{T_u}(v_i, \alpha) = \begin{cases} \Delta^{T_u}(v_i) & \textit{leaf}(v_i)\textit{=true} \\ \rho^{T_u}(v_i, \alpha) & \textit{root}(v_i)\textit{=true} \\ \Phi^{T_u}(v_i, \alpha) & \text{Otherwise} \end{cases} \quad (4)$$

  This function considers nodes in a multi-tree in three categories: leaves, root and nodes situated between leaves and the root whose weights are calculated by functions Δ, ρ and Φ respectively.

- Δ is a function mapping $V \rightarrow \{0, 1\}$. This function determines whether a specific node, v_i, in a combined tree T_u exists in both of the trees from which it is constituted (T_1, T_2).

$$\Delta^{T_u}(v_i) = \begin{cases} 1 & \text{if } v_i \in V^{T_1}, v_i \in V^{T_2} \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

- ρ is a function mapping $V \times \mathbb{R} \rightarrow \mathbb{R}$. This function is used to calculate the weights of the nodes in a multi-tree.

$$\rho^{T_u}(v_i, \alpha) = (\frac{1}{\beta_{v_i}})(\sum_{\forall v_x \in children(v_i)} P(v_i, v_x) W^{T_u}(v_x, \alpha)) \quad (6)$$

- Φ is a function mapping $V \times \mathbb{R} \rightarrow \mathbb{R}$. This function returns the weight of a node if the node is neither a leaf node nor the root of the multi-tree.

$$\Phi^{T_u}(v_i, \alpha) = (1 - \frac{1}{\alpha^{L(v_i)+1}}) \rho^{T_u}(v_i, \alpha) \quad (7)$$

$$+ (\frac{1}{\alpha^{L(v_i)+1}}) \Delta^{T_u}(v_i)$$

  The Φ function calculates the weight of a node by $1 - \frac{1}{\alpha^{L(v_i)}}$ share of the average of the weight of its children which calls the function to calculate the weight of each child recursively and $\frac{1}{\alpha^{L(v_i)}}$ share for the commonality of a node between the multi-trees of two concepts.

The above description is best illustrated by the following example. The example, shown in Figure 1,2 and the calculation using equations 5-7 illustrated in Figure 3, demonstrates how the above functions work for two entities represented by their features (i.e., products appeared in the profiles of two users).
d_1=(Web Cam, Digital Camera, LCD TV, Blender, Mattress)
d_2=(Keyboard, DSLR Camera, LED TV, Mattress, Drawer)

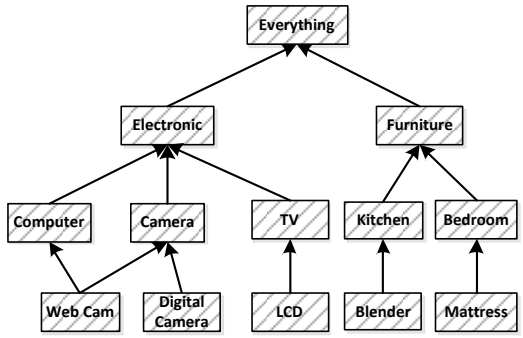

Figure 1: First multi-tree representing transaction d_1

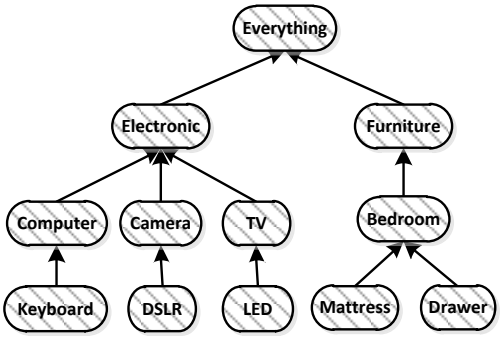

Figure 2: Second multi-tree representing transaction d_2

Using a VSM, the similarity between d_1, d_2 is equal to 0.2. However, using the proposed method, although LED and LCD are not equal they have a parent in common which makes the weight non-zero. Considering $\alpha = e = 2.71$ (also used in experiments reported later), the similarity between d_1, d_2 is: 0.444.
W(Keyboard)=0, W(Web Cam)=0, W(Digital Camera)=0, W(DSLR)=0, W(LED)=0, W(LCD)=0, W(Blender)=0, W(Drawer)=0, W(Mattress)=1, W(Kitchen)=0
W(Computer)=W(TV)=W(Camera)=$(1 - \frac{1}{e})(0) + (\frac{1}{e}) = 0.369$
W(Bed room)=$(\frac{1}{2}) \times (1 - \frac{1}{e}) + (\frac{1}{e}) = 0.684$
W(Electronic)=$(\frac{1}{e}) \times (1 - \frac{1}{e^2}) + (\frac{1}{e^2}) = 0.457$

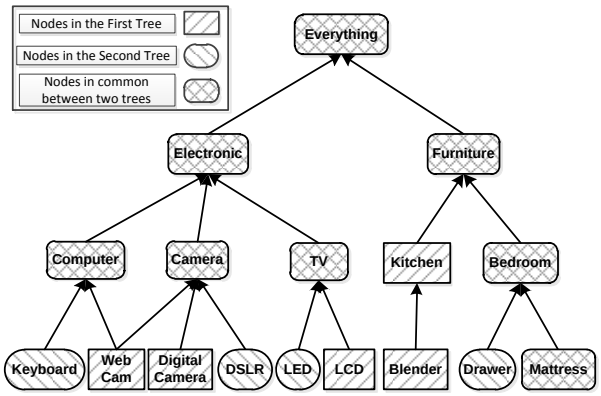

Figure 3: A multi-tree combined from previous two multi-trees

W(Furniture)=$(\frac{0+0.684}{2}) \times (1-\frac{1}{e^2}) + (\frac{1}{e^2}) = 0.431$
W(Everything)=0.444

4 Similarity using a Semantic Multi-Tree

In order to calculate the similarity between two entities, we construct two multi-trees each of which represents features of the corresponding entity in a hierarchical format according to a specific ontology or taxonomy. In this method, each entity is represented by its features as leaves of a multi-tree. The rest of each multi-tree is constructed according to the domain taxonomy (e.g., WordNet or Wikipedia Categories). Hence, the multi-tree corresponding to each entity is a sub multi-tree of the taxonomy with which the sub-multi-tree is constructed. A multi-tree T_x is said to be a sub multi-tree of $T_\mathcal{O}$ if:

$$T_x \subseteq T_\mathcal{O} \Leftrightarrow \begin{cases} V_x \subseteq V_\mathcal{O} \\ E_x \subseteq E_\mathcal{O} \\ C_x \subseteq C_\mathcal{O} \end{cases} \quad (8)$$

Assume that, $T_\mathcal{O} = (V_\mathcal{O}, E_\mathcal{O}, C_\mathcal{O}, L_\mathcal{O}, M_\mathcal{O}, W_\mathcal{O}, P_\mathcal{O})$, is a multi-tree representing the domain taxonomy (e.g., Wikipedia Categories), $\mathcal{O} = (\mathcal{C}_\mathcal{O}, \mathcal{R}_\mathcal{O})$, in which $\mathcal{C}_\mathcal{O}$ represents set of concepts and $\mathcal{R}_\mathcal{O}$ represents set of relations among concepts in the taxonomy. The transformation function $\mathcal{T}$ is defined as a bijective function $\mathcal{T} : \mathcal{R}_\mathcal{O} \rightarrow E$ which maps each relation in the taxonomy $\mathcal{O}$ to an edge in the multi-tree $T_\mathcal{O}$. So, $E_x = \{e_i = \mathcal{T}(R_i) \mid R_i \in \mathcal{R}_\mathcal{O}\}$. ($C_\mathcal{O} \equiv \mathcal{C}_\mathcal{O}$ and $E_\mathcal{O} \equiv \mathcal{R}_\mathcal{O}$).

The multi-tree, $T_x = (V_x, E_x, C_x, L_x, M_x, W_x, P_x) \subseteq T_\mathcal{O}$, corresponds to entity $d_x = (c_1, ..., c_n)$ in which c_i is a term representing a feature of this entity as well as a concept in the taxonomy. We define $C_x \subseteq C_\mathcal{O}$ in multi-tree T_x as a set of terms representing features of the entity d_x.

Hence, $C_x = \{c_1, ..., c_n\} \cup \{c_j \mid \forall c_i \in C_x, \forall c_j \in \mathcal{C}_\mathcal{O}, \textit{is-a}(c_i, c_j) \in \mathcal{R}_\mathcal{O}\}$, $V_x = \{v_i = M(t_i) \mid t_i \in C_x\}$ and $E_x = \{e_i = \mathcal{T}(R_i) \mid \forall c_k, c_l \in C_x, R_i(c_k, c_l) \in R_\mathcal{O}\}$ such that $C_x \subseteq \mathcal{C}_\mathcal{O}$ and $\mathcal{O} = (\mathcal{C}_\mathcal{O}, \mathcal{R}_\mathcal{O})$ is the taxonomy which is used to construct the tree.

In the next step, the combination operator is applied to the two trees whose similarity is being computed. Applying the combination operator to the two trees, the weight of the root of the combined tree represents the similarity of the two trees. The weight of the root is recursively calculated by application of equations 5-7. The following steps demonstrate the process of calculating the similarity between two entities d_1, d_2 represented by sets of features C_1, C_2 respectively:

1. Construct multi-trees T_1 and T_2 from sets of features C_1 and C_2 respectively.
2. Construct $T_{sim} = T_1 \uplus T_2$ as a combination of two multi-trees $T_1, T_2 \subseteq T_\mathcal{O}$
3. Update the weights for the nodes in the combined multi-tree T_{sim} using the recursive equations 5-7.
4. The weight of the root of T_{sim} is the value which represents the similarity of two entities represented by C_1, C_2; i.e., $Sim(d_1, d_2) = W(root(T_{sim}))$.

Algorithm 1 and 2 describes the process by which a multi-tree is constructed from a feature set representing an entity.

Algorithm 1 Constructing a multi-tree.

Proc ConstructMulti-Tree(ConceptSet C_x)
$T_x \leftarrow null$
for all c in C_x **do**
  FindPaths($T_\mathcal{O}.M^{-1}(c), T_\mathcal{O}, T_x$)
end for
return T_x

Algorithm 2 Finding a path in a multi-tree from a leaf node to root.

Proc FindPaths(Node v,Multi-Tree $T_\mathcal{O}$,Multi-Tree T_x)
if root(v) **then**
  $T_x.root \leftarrow v$
  return
end if
for all parent in $T_\mathcal{O}$.Parents(v) **do**
  if parent not in $T_x.V$ **then**
    $T_x.V \leftarrow T_x.V \cup parent$
    $T_x.E \leftarrow T_x.E \cup < parent, v >$
    FindPaths(parent,$T_\mathcal{O}, T_x$)
  end if{avoid cycles}
end for

Algorithm 3 describes the calculation of similarity using the proposed non-linear method.

Algorithm 3 Calculation of similarity using multi-trees.

$T_1 \leftarrow$**ConstructMulti-Tree**(ConceptSet C_1)
$T_2 \leftarrow$**ConstructMulti-Tree**(ConceptSet C_2)
$T_{Sim} \leftarrow T_1 \uplus T_2$
$similarity \leftarrow W^{T_{Sim}}(root, \alpha)$

5 Experimental Results

The proposed model is not only useful for measuring similarity between pairs of words but is also useful for information retrieval and recommender systems. In this paper, we evaluated the semantic multi-tree model for the application of measuring similarity between pairs of words, but the domain of the proposed model is not limited just to the application of measuring similarity between pairs of words. Our rationale for doing this is that the concept domain is potentially larger as we are not limited to (say) movies, music or books, domains often used in recommender datasets.

One of the methods for evaluating psycholinguistic systems is comparing the results with human judgement. Among three standard datasets that exist in the domain of measuring semantic relatedness between pairs of words, WordSimilarity-353 is the most comprehensive dataset since this dataset includes all of the 30 nouns of the Miller and Charles dataset and most of the 65 pairs of Rubenstein and Goodenough testset [Rubenstein and Goodenough, 1965]. The WordSimilarity-353 dataset contains 353 pairs of words compared by human agents in terms of similarity [Finkelstein *et al.*, 2002]. This system has been evaluated by both WordSimilarity-353 and Rubenstein and Goodenough testsets. While the Charles and Miller testset is not available on the web and is already included in WordSimilarity-353, we are not able to give statistics about the performance of proposed method on this dataset.

Wikipedia is a resource of concepts linked to each other forming a network, which is collaboratively constructed by human agents around the world. Each page in Wikipedia is linked to a set of categories classifying concepts and Wikipedia pages. Each page in Wikipedia describes one of the concepts associated to a word. A Wikipedia page describes a concept using other concepts described in other pages. In order to evaluate the proposed model to estimate the similarity between two entities, we used Wikipedia as a resource of knowledge and Wikipedia Categories as a taxonomy of concepts with which Wikipedia pages are annotated. The existing links in a Wikipedia page are considered as features describing the associated concept. Figure 4 illustrates the Wikipedia link structure data model. In this figure, each circle represents a wikipedia category and each square represents a Wikipedia page corresponding to a concept. Solid lines represent links between pages and a dotted line represents a link between a page and categories it belongs to.

In this experiment, The VSM is compared to the multi-tree model described previously against human judgement in terms of accuracy and correlation. For this purpose, each word is mapped to a Wikipedia page, and then a vector containing the links of the pages to which the corresponding page is linked is constructed and is referred to as a *link vector*. In Figure 4, the page L is linked to $\{N, O, P\}$ which are considered as features of the link vector $C_L = (N, O, P)$. Each link in the link vector represents another page in Wikipedia which is linked to Wikipedia categories. In the next step, another vector is constructed according to the categories of a link vector's elements (i.e., leaf nodes in Categories) called a first order category vector. In Figure 4, $\{N, O, P\}$ are

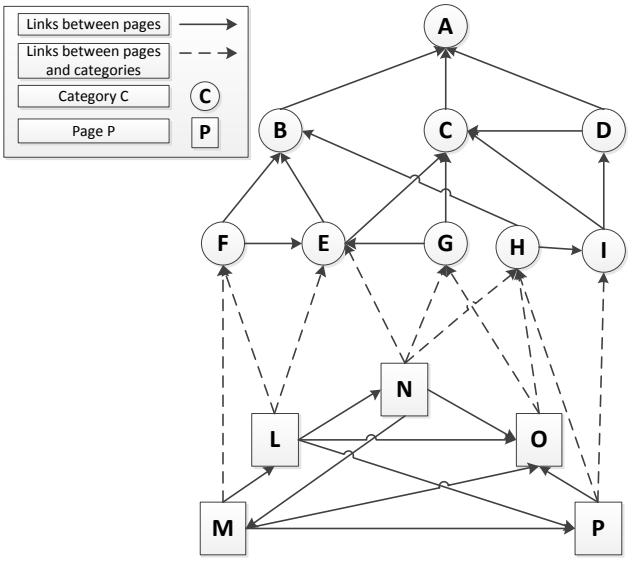

Figure 4: The Wikipedia link structure data model

linked to categories $\{E, G, H, I\}$. Then, the categories of the main page, L, (i.e., $\{E, F\}$) are added to the first order category vector and then the multi-tree representing the concept L is constructed from the first order category vector ($C_L^{1st} = (E, F, G, H, I)$). The rest of the process for construction of the multi-tree model is the same as described in Sections 3 and 4. This process is pictorially represented in Figure 5.

In VSM, the link vectors are compared using cosine similarity as described in equation 1. Both VSM schemes have been evaluated against human judgement with the same dataset and the results are compared in Table 1. Since, in this paper, we are not using a corpus-based approach, and in each experiment only two vectors representing two concepts are compared, the inverse document frequency of each link can not be calculated. Therefore, tf was used instead of tf-idf to implement the second VSM scheme.

The average accuracy in Table 1 is measured regarding the difference between the value of similarity measured by the techniques tested above and that of human judgement.

$$Accuracy = 1 - Average(error_i) \tag{9}$$

$$error_i = Sim_{Human}(w_{i1}, w_{i2}) - Sim_{Computer}(w_{i1}, w_{i2}) \tag{10}$$

The correlation was estimated by Spearman's rank correlation coefficient. The correlation between human judgement and the multi-tree model is demonstrated in Figure 6. Table 1 demonstrates that the proposed model achieved better results than both VSM schemes in terms of accuracy and correlation with human judgment in estimating similarity between entities.

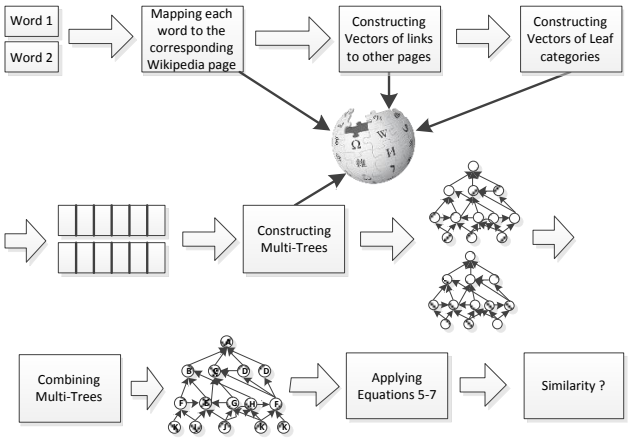

Figure 5: The Architecture of the proposed system

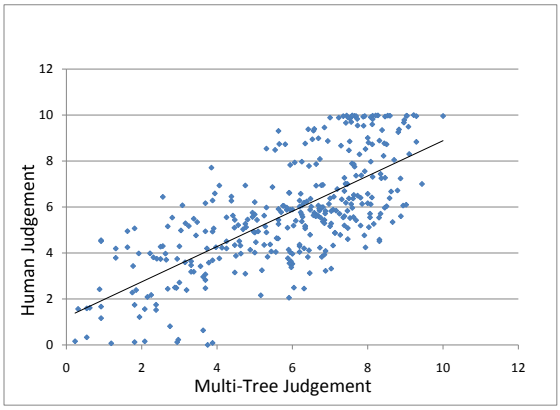

Figure 6: Human Judgement vs. Multi-Tree similarity algorithm on WordSimilarity-353 dataset

6 Discussion, Conclusions and Future Work

In this paper we observed that techniques such as linear VSM ignore the semantic relationships among features. VSM calculates the similarity between two documents regarding the commonality of their features. However, in some cases, two documents may not be equal but may refer to the same entity. This limits the capability of a VSM to retrieve related documents. The multi-tree model compensates for the lack of semantic relatedness among features using taxonomic relations that exist among the features of two entities. In this model the similarity weight is propagated from leaf nodes to the root of the multi-tree. The multi-tree model was evaluated by using the WordSimilarity-353 and Rubenstein and Goodenough datasets against human judgement and the results show that the multi-tree method outperforms VSM in terms of correlation and the average accuracy against human judgement for similarity of pairs of words. The results for WikiRelate and WLM, two previous systems which used Wikipedia Categories or WordNet to perform the task of similarity between two words using the same dataset, are shown in Table 2. WikiRelate and WLM were briefly described in Section 2.

	Average Accuracy compared to human judgement	Correlation with Human judgement
WordSim-353		
VSM Boolean	57.1	54
VSM Frequency of terms	59.2	56.9
Multi-Tree	84.7	70.6
Rubenstein and Goodenough		
VSM Boolean	61.9	57.1
VSM Frequency of terms	62.2	60
Multi-Tree	80	74

Table 1: The comparison between VSM and Multi-Tree model using two testsets.

The results in Table 2 show that the method proposed in this paper outperforms two of the other competitors namely WikiRelate and WLM by more than 20% and 3% respectively. However, ESA is the first ranked system using machine learning techniques as the basis of a semantic interpreter that is part of a system that maps fragments of natural language text into a weighted sequence of Wikipedia concepts ordered by their relevance to the input. Regarding ESA, it is clear that augmenting the text representation and constructing a weighted list of concepts provides benefits that link analysis alone does not completely replace.

Dataset	WikiRelate	ESA	WLM	Multi-Tree
Goodenough	52	82	64	74
WordSim-353	49	75	69	71
W-average	49	76	68	71

Table 2: Performance of semantic relatedness measures for two standard datasets for three popular systems vs. Multi-tree model.

Therefore, while the multi-tree model shows promise as a means by which semantically similar entities can be found, there are various ways that we can extend this approach. For instance, in this model we ignored the number of occurrences of features in each multi-tree as initial weights for leaves and used a binary scheme to calculate node weight.

Another potential model extension is in using a more sophisticated function such as Pearsons product-moment correlation or cosine similarity instead of the simple average function in equation 6.

A potential application of this model is in recommender systems, which concentrate on similarity between two products or two people. Referring once again to the example described in Section 3 and shown graphically in Figures 1, 2 and 3, the similarity of buying patterns can be established

using a semantic multi-tree approach. For the evaluation of such systems, we need to construct a handcrafted taxonomy of products plus annotation of the product dataset to the taxonomy of products. Keyword search engines are also another potential application of such systems instead of linear VSM.

References

[Annates, 2005] U. Annates. Semantic tree method-historical perspective and applications Izabela Bondecka-Krzykowska. *Annales Universitatis Mariae Curie-Skłodowska: Informatica*, page 15, 2005.

[Finkelstein *et al.*, 2002] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM Transactions on Information Systems (TOIS)*, 20(1):116–131, 2002.

[Gabrilovich and Markovitch, 2007] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 6–12, 2007.

[Landauer *et al.*, 1998] T.K. Landauer, P.W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2):259–284, 1998.

[Larkey and Markman, 2005] L.B. Larkey and A.B. Markman. Processes of similarity judgment. *Cognitive Science: A Multidisciplinary Journal*, 29(6):1061–1076, 2005.

[Pedersen *et al.*, 2005] S.P.T. Pedersen, S. Banerjee, and S. Patwardhan. Maximizing semantic relatedness to perform word sense disambiguation, 2005. *University of Minnesota Supercomputing Institute Research Report UMSI*, 25, 2005.

[Rubenstein and Goodenough, 1965] H. Rubenstein and J.B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[Salton and McGill, 1983] G. Salton and M.J. McGill. Introduction to modern information retrieval. *New York*, 1983.

[Salton *et al.*, 1975] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[Strube and Ponzetto, 2006] M. Strube and S.P. Ponzetto. WikiRelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1419. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[Tversky and Shafir, 2004] A. Tversky and E. Shafir. *Preference, belief, and similarity: selected writings*. The MIT Press, 2004.

[Witten and Milne, 2008] I.H. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.