Measuring Semantic Similarity using a Multi-Tree Model

Behnam Hajian and Tony White

School of Computer Science Carleton University

Jan, 2011

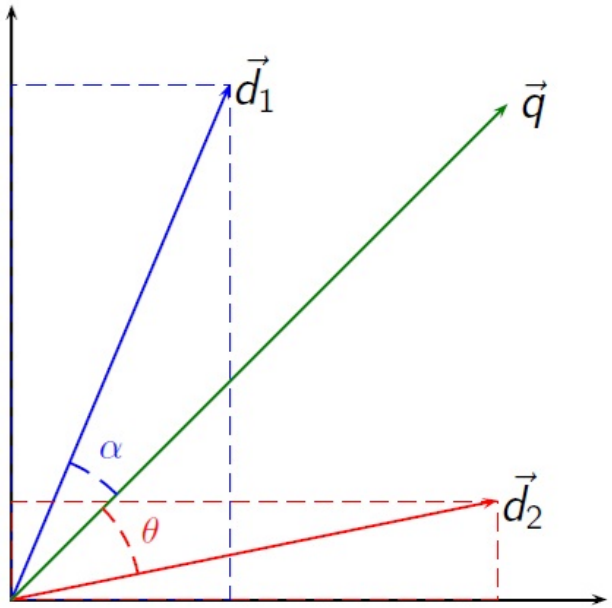
What is Similarity?

- **Similarity** refers to psychological nearness between two concepts.
- **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).
- Since each concept is represented by the features describing its properties, a similarity comparison involves comparing the feature lists representing that concept.
- Potential applications:
  - recommendation systems, e-commerce,
  - search engines, biomedical informatics
  - natural language processing tasks such as word sense disambiguation.

Vector Space Model

- In the Vector Space Model (VSM), a document or a query is represented as a vector of identifiers such as index terms.
  $d_j = (w_{1j}, ..., w_{nj}$
- An example:
  - 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.



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

$$= \frac{\sum_{k=1}^{N} w_{ik} \times w_{qk}}{\sqrt{\sum_{k=1}^{N} w_{ik}^2} \cdot \sqrt{\sum_{k=1}^{N} w_{qk}^2}}$$

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

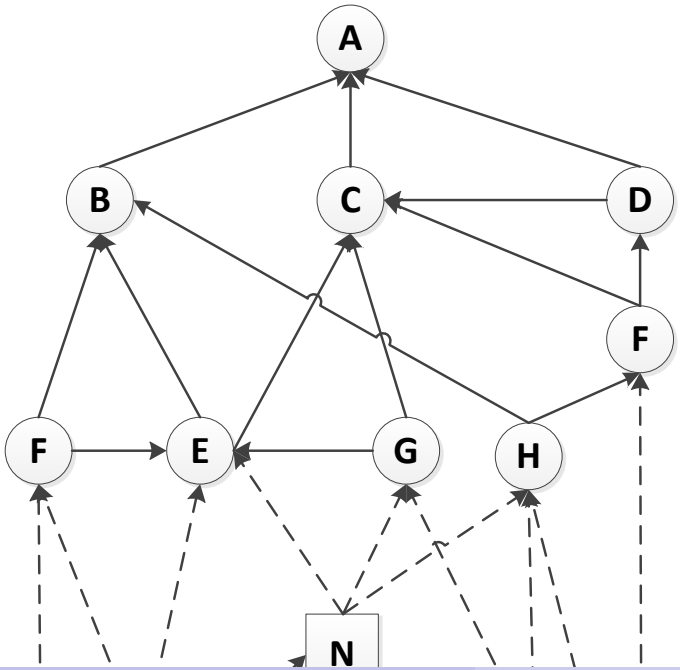
Related Works

several approaches for measuring semantic relatedness using Ontologies (e.g, WordNet, Wikipedia Categories)

- WikiRelate (Strube and Ponzetto 2006),
- Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch 2007)
- Wikipedia Link-based Measure (WLM) (Witten and Milne 2008).

The Definition for Taxonomy and Multi-Tree

- **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.



Multi-Tree (Multiple Parent Tree)- Formal Definition

A Multi-Tree or Multiple Parent Tree is a Digraph (directed graph) $T = (V, E, C, L, M, W, P)$ with hierarchical order of its node in different levels.
in which:

- 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. Each edge represents an *is-a* relation between two concepts c_i, c_j which means (c_i *is-a* c_j).
- 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.

Multi-Tree (Multiple Parent Tree)- Formal Definition (cont.)

Regarding $T = (V, E, C, L, M, W, P)$:

- 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.
- 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.

Multi-Tree (Multiple Parent Tree)- Formal Definition (cont.)

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 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 parents of the node.

Multi-Tree (Multiple Parent Tree)- Formal Definition (cont.)

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

- $\beta_v = |children(v)|$ is defined as the cardinality of the child set of node v.
- $\gamma_v = |parents(v)|$ is defined as the cardinality of the parent set of 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

$$T_u = T_1 \uplus T_2 \Rightarrow T_u = \begin{cases} E_u = E_1 \cup E_2 \\ V_u = V_1 \cup V_2 \\ C_u = C_1 \cup C_2 \end{cases} \begin{cases} L^{T_u} \\ M^{T_u} \\ P^{T_u} \\ W^{T_u} \end{cases} \quad (1)$$

Multi-Tree (Multiple Parent Tree)- Formal Definition (cont.)

- The weights of the vertices in the tree T_u are calculated by a recursive function $W^{T_u} : V \times \mathbb{R} \rightarrow \mathbb{R}$. α is a damping factor (degradation ratio).

$$W^{T_u}(v_i, \alpha) = \begin{cases} \Delta^{T_u}(v_i) & \text{Leaf}(v_i)\text{=true} \\ \rho^{T_u}(v_i, \alpha) & \text{Root}(v_i)\text{=true} \\ \Phi^{T_u}(v_i, \alpha) & \text{Otherwise} \end{cases} \quad (2)$$

- 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).

$$\Delta^{T_u}(v_i) = \begin{cases} 1 & \text{if } v_i \in V^{T_1}, v_i \in V^{T_2} \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

Multi-Tree (Multiple Parent Tree)- Formal Definition (cont.)

- ρ is a function mapping $V \times \mathbb{R} \rightarrow \mathbb{R}$. This function returns the weight of a node if the node is the root of the multi-tree.

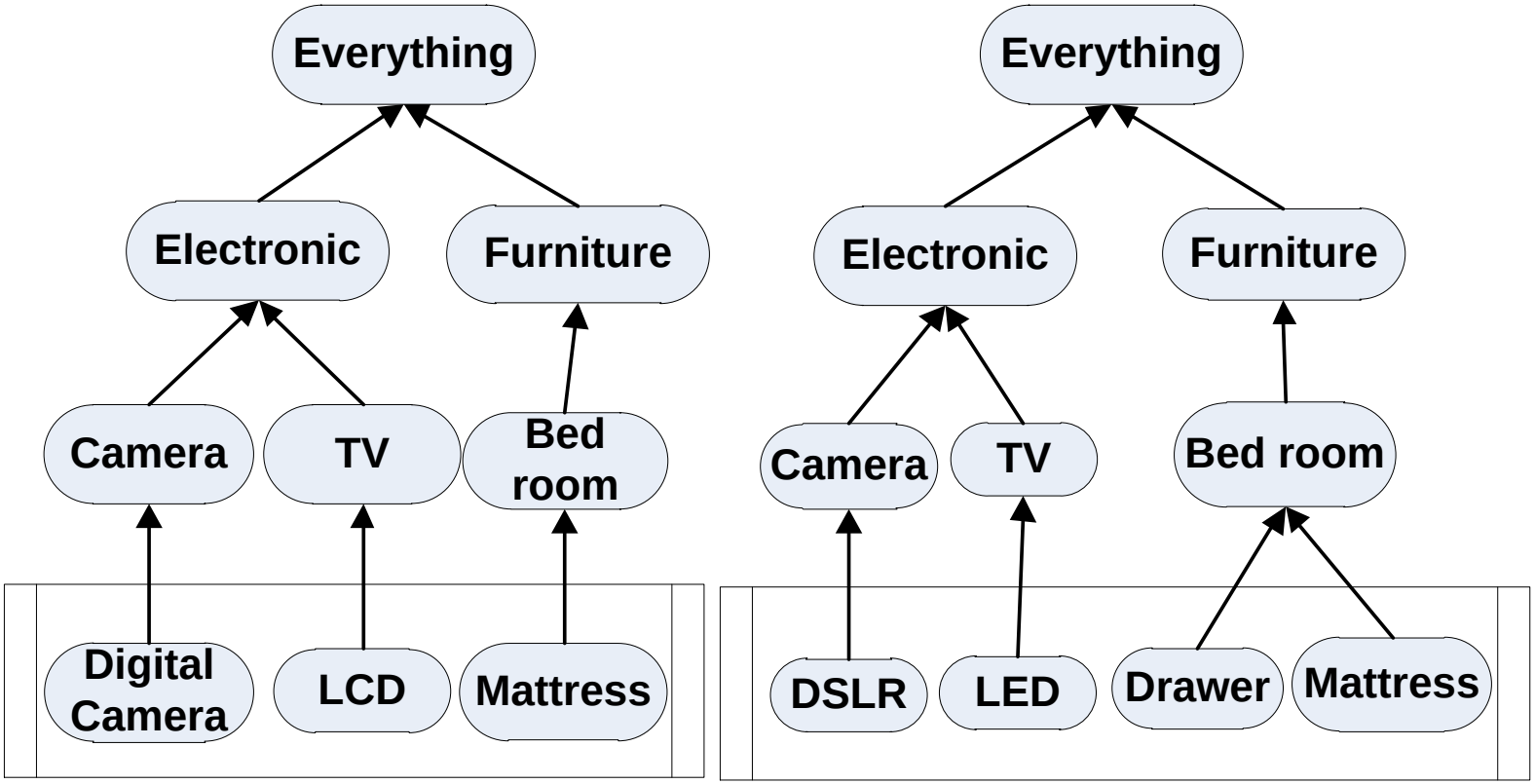
$$\rho^{T_u}(v_i, \alpha) = (\frac{1}{\beta_{v_i}})(\sum_{\forall v_x \in children(v_i)} P(v_i, v_x) W^{T_u}(v_x, \alpha)) \quad (4)$$

- 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.

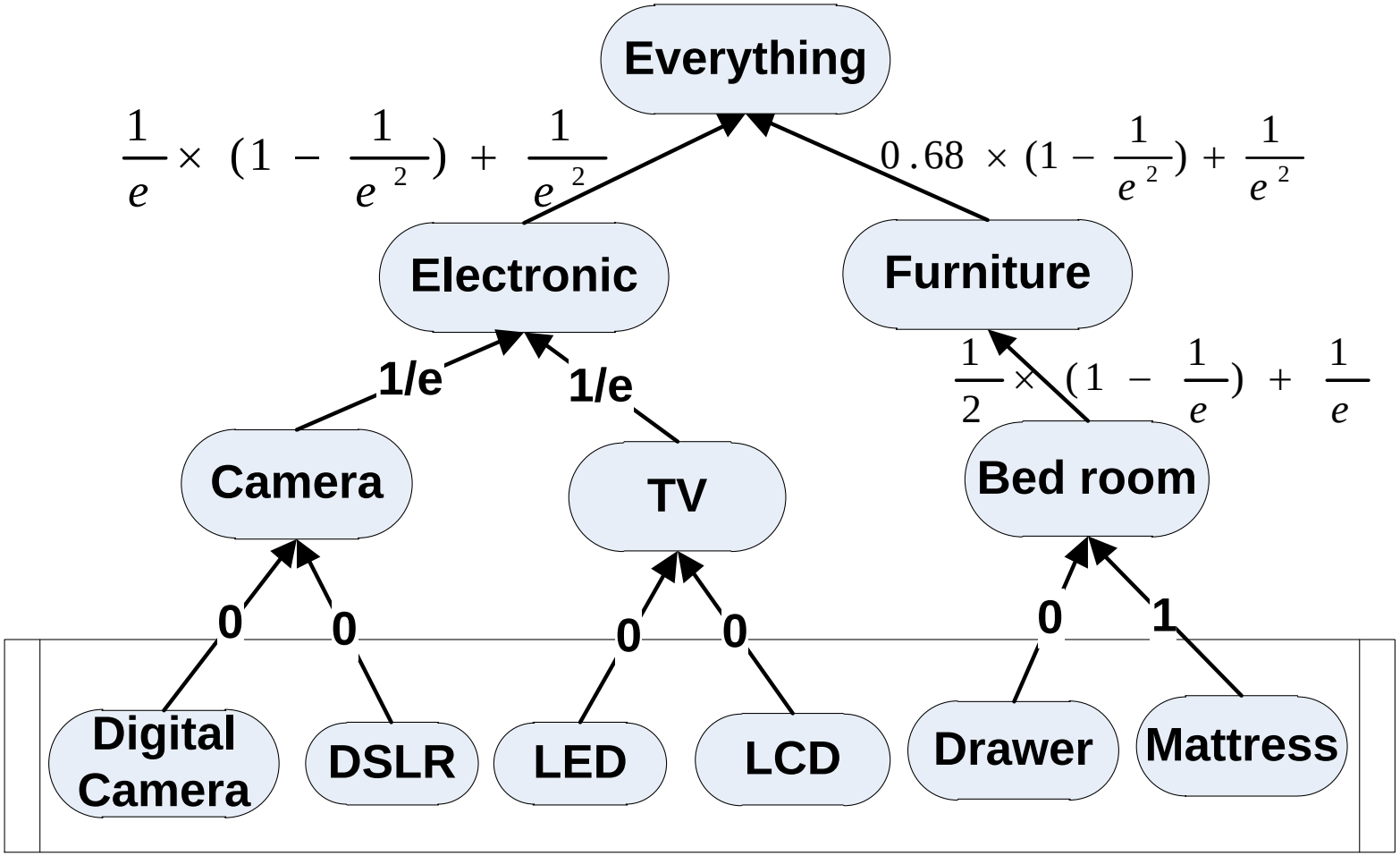
$$\Phi^{T_u}(v_i, \alpha) = (1 - \frac{1}{\alpha^{L(v_i)+1}})\rho^{T_u}(v_i, \alpha) \quad (5)$$

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

Example:


Everything
Electronic
Furniture
Camera
TV
Bed room
Digital Camera
LCD
Mattress
Everything
Electronic
Furniture
Camera
TV
Bed room
DSLR
LED
Drawer
Mattress

Example:


Everything
$\frac{1}{e} \times (1 - \frac{1}{e^2}) + \frac{1}{e^2}$
$0.68 \times (1 - \frac{1}{e^2}) + \frac{1}{e^2}$
Electronic
Furniture
1/e
1/e
$\frac{1}{2} \times (1 - \frac{1}{e}) + \frac{1}{e}$
Camera
TV
Bed room
0
0
0
0
0
1
Digital Camera
DSLR
LED
LCD
Drawer
Mattress

Example:

$$\text{Weight(Camera)}=(1-\tfrac{1}{e})(0)+(\tfrac{1}{e})$$
$$\text{Weight(TV)}=(1-1/e)(0)+(1/e)$$
$$\text{Weight(Bed room)}=(\frac{1}{2})\times(1-\tfrac{1}{e})+(\tfrac{1}{e})=0.684$$
$$\text{Weight(Electronic)}=(\tfrac{1}{e})\times(1-\tfrac{1}{e^2})+(\tfrac{1}{e^2})=0.457$$
$$\text{Weight(Furniture)}=(0.684)\times(1-\tfrac{1}{e^2})+(\tfrac{1}{e^2})=0.723$$
$$\text{Weight(Everything)}=0.59$$

Constructing a Multi-Tree from a Taxonomy:

Assuming $T_{\mathcal{O}} = (V_{\mathcal{O}}, E_{\mathcal{O}}, C_{\mathcal{O}}, L_{\mathcal{O}}, M_{\mathcal{O}}, W_{\mathcal{O}}, P_{\mathcal{O}})$, as a multi-tree representing the domain taxonomy $\mathcal{O} = (\mathcal{C}_{\mathcal{O}}, \mathcal{R}_{\mathcal{O}})$,

- The transformation function $\mathcal{T}$ is defined as a bijective function $\mathcal{T} : \mathcal{R}_{\mathcal{O}} \rightarrow E$ mapping each relation in the taxonomy $\mathcal{O}$ to an edge in the multi-tree $T_{\mathcal{O}}$. So, $E_x = \{e_i = \mathcal{T}(R_i) \mid R_i \in \mathcal{R}_{\mathcal{O}}\}$. $(C_{\mathcal{O}} \equiv \mathcal{C}_{\mathcal{O}}$.
- 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.
- $C_x = \{c_1, ..., c_n\} \cup \{c_j \mid \forall c_i \in C_x, \forall c_j \in \mathcal{C}_{\mathcal{O}}, \textit{is-a}(c_i, c_j) \in \mathcal{R}_{\mathcal{O}}\}$, $V_x = \{v_i = M(t_i) \mid t_i \in C_x\}$ 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 $C_x \subseteq \mathcal{C}_{\mathcal{O}}$

The process of calculating the similarity between two entities:

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 1-3.
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., $Sim(d_1, d_2) = W(Root(T_{sim}))$.

Constructing Multi-Tree by Finding all the paths from leafs to root

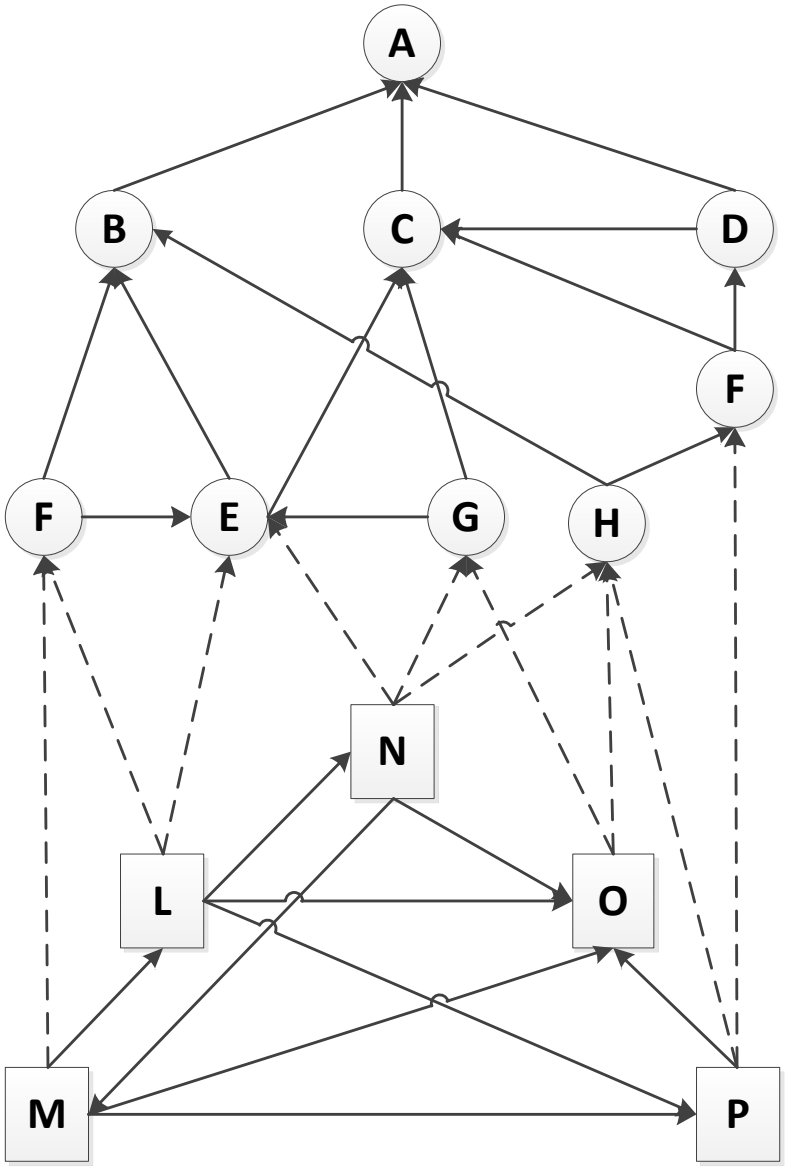
Proc ConstructMulti-Tree(ConceptSet C_x)
$T_x \leftarrow null$
for all c in C_x **do**
  FindPaths($T_{\mathcal{O}}.M^{-1}(c), T_{\mathcal{O}}, T_x$)
end for
return T_x

Constructing Multi-Tree by Finding all the paths from leafs 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**
  $T_x.V \leftarrow T_x.V \cup parent$
  $T_x.E \leftarrow T_x.E \cup < parent, v >$
  FindPaths(parent, $T_{\mathcal{O}}$, T_x)
end for

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

Experimental Results



- Wikipedia is a resource of concepts linked to each other forming a network, which is collaboratively constructed by human agents around the world.
- Wikipedia Categories as a taxonomy of concepts to whichWikipedia pages are annotated.
- The WordSimilarity-353 dataset contains 353 pairs of words

Experimental Results

	Average Accuracy compared to human judgment	Correlation with Human results
VSM Boolean	54.7	55.9
VSM Frequency of terms	55.7	52.9
Multi-Tree	80.9	72.8

Table: The comparison between VSM and Multi-Tree model using WordSimilarity-353.

Conclusions

1. 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.
2. The multi-tree model compensates for the lack of semantic relatedness among features using taxonomic relations that exist among the features of two entities.

Future work

1. 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.
2. using a more sophisticated function such as Pearsons product-moment correlation or cosine similarity instead of the simple average function in equation
3. Another potential application of this model is in recommender systems, which concentrate on similarity between two products or two people. 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.
4. Keyword search engines are also another potential application of such systems instead of linear VSM.

References

1. Annates, U. 2005. Semantic tree method-historical perspective and applications Izabela Bondecka-Krzykowska. Annales Universitatis Mariae Curie-Sklodowska: Informatica 15.
2. Pedersen, S.; Banerjee, S.; and Patwardhan, S. 2005. Maximizing semantic relatedness to perform word sense disambiguation, 2005. University of Minnesota Supercomputing Institute Research Report UMSI 25.
3. Salton, G., and McGill, M. 1983. Introduction to modern information retrieval. New York.
4. Strube, M., and Ponzetto, S. 2006. WikiRelate! Computing semantic relatedness usingWikipedia. In Proceedings of the National Conference on Artificial Intelligence, volume 21, 1419. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
5. Tversky, A., and Shafir, E. 2004. Preference, belief, and similarity: selected writings. The MIT Press.