Clustered Hierarchical Concept Based Semantic Closeness Between Two Concepts Using WordNet

Boddu Bhaskara Rao[1], Vatsavayi Valli Kumari[2]

[1] Department of Information Technology, GIT, GITAM University
Visakhapatnam, Andhra Pradesh-45, India

[2] Department of CSSE, College of Engineering, Andhra University
Visakhapatnam, Andhra Pradesh-03, India

Abstract

The search engine needs relatedness to measure closeness between two concepts for determining optimal results in major applications like information retrieval, information integration and of many more in natural language processing tasks i.e. text classification, word sense disambiguation, matching problems in artificial intelligence etc,. The clustered hierarchical concept network helps to overcome the fuzzy variations in different levels of granularity in measures of closeness based on weights, frequency or distances but these measures are not considered since no method takes the actual context of the user intention, user query or context domain subject fields. Clustered hierarchical concept network has three steps: Elicitation: extract the concepts of user query using concept extraction algorithm and name the output as context domain. Construction: building hierarchical clusters based on context or concept domain with related concepts as nodes and relations as edges. Matching: determine the matching concepts like Least Common General Concept (LCGC) and Least Common Specific Concept (LCSC). Clustered hierarchical concept based semantic closeness has three features i.e., context domain, concept net and common concepts. These features are used to calculate the relatedness. The primary goal of hierarchical concept network is to include the semantic of the concept by including its three features. The extraction of concepts are not only related to individual concepts, but it is also an organizational structure of the concepts that are combined in the ontology i.e. WorNet. In this paper, we propose a method for computing semantic closeness of two concepts in which the holonyms, meronyms, instances of concepts are considered synthetically. By calculating test data, the experiment results show that the method can compute concepts closeness effectively. The human judgments on a set of concept pairs led our approach to be more effective and have shown one of the best performance than the measures based on concept vector.

Keywords: *Search Engine, Word Sense, Relatedness, Clustered Hierarchical Concept, Elicitation, Construction, Matching, LCGC, LCSC, WordNet, Holonyms, Meronyms and Instances.*

1. Introduction

The powers of polysemy and synonymy that exist in WordNet [1] of natural language have become a challenge in the many applications of Natural Language Processing (NLP) and Information Retrieval (IR) [2 3]. The humans have little difficulty in determining the exact meaning of ambiguous concepts, while to automate the process of replication. The system has to calculate the semantic closeness before finalizing the output of above applications. It is a primary tool not only for NPL but also many applications like, example-based machine translation, Information retrieval, Information Integration, text classification and many more. All these applications fully depend on the requirement of measure of appropriate semantic closeness between the concepts. We use ontology integration and retrieval core techniques for achieving good results in Information retrieval, integration and many more applications of NLP tasks.

A variety of models have been developed based on the degree of overlap [4] between the concepts. In general, elicitation of semantic similarity [5] of a concept is a big issue in overlap technique also. Artificial Intelligence aims to find techniques or procedure to process and organize the concepts in such way that the reasoning methods can get semantics efficiently. One such method is our clustered hierarchical concept network using WordNet[1] for semantic similarity or closeness of two concepts. Here the two concepts create two clusters. Each one has its own features. It is a concept network [6], in which nodes are concepts and edges are the hierarchical relations between the two adjacent nodes that connect. The definition of a concept C_i is defined as a union of three tuple (ID_i, SYS_i, CTX_i) where ID_i indicates unique lexical unit or word or term of the concept C_i, SYS_i indicates set of terms that gives same concept of C_i and CTX_i indicates terms that are adopted by the concept C_i i.e. instances, hyponyms and hypernyms. When a word level semantic relation requires exploration, it has many potential types of relations that can be considered. Hierarchical relation means hypernym-hyponym, part-whole and equivalence i.e. synonymy.

Among all these, the hierarchical relation represents a major and the most important role in this work. To get good results, the construction of concept network should be done efficiently and effectively. This is essential in most of the applications of natural language processing and information retrieval. Clustered hierarchical concept network based on semantic closeness combines both path-based and information context measures. Individually each one has its own advantages and disadvantages, yields good and unexpected results. We combine advantages of both techniques to overcome the limitations of both the measures by forming concept clusters and concept network [6]. For example, take two concepts: lecturer and teacher. Humans can easily recognize that these concepts carry the same thing and are related to educational institutions. Here there is no overlapping of one concept with the other. We can determine the correct sense of ambiguity from its context based on the interrelationships and dependencies between the concepts. The machine can't do this unless the computer is really as smart as human brain. There is no strict definition of similarity because of lack of objective criteria, but subjectively it is strong. That's why the humans have consensus of opinions if two concepts are similar in some occasions under certain scenarios.

Semantic relatedness will play vital role in most of the applications such as sense disambiguation, concept narratives in the field of NLP and IR that we have specified earlier. Our objective is to automatically measure the closeness of two concepts as same as human. Our measure is based on clusters, hierarchical relations and wordNet as ontology. It evaluates the semantic similarity of concepts from the semantic information, semantic relations in the wordNet as ontology. It has given good accuracy when compared to other measures like concept vector with respect to the bench mark of human judgment.

2. Related Works

No strict definition of semantic similarity makes the natural language processing complex and complicated. A lot of work has been done in this field. The summarized work has been discussed in this section. In [7], Hishan Al-Mubaid and Nguyen presented a cross-cluster method for measuring the semantic distance between two concepts using wordNet [1]. This method overcomes the differences of granularity level of clusters in wordNet. They have defined three features based on which the semantic distance is measured. In case of polysemes of concepts, the measure gives same result every time since it is not considering the polysemes of the concepts during the calculation of semantic similarity. The determination of common specificity feature follows one type of relation on two concepts. This avoids other senses of the concepts. In [8], Ahmad EI Sayed et al, presented a new context-aware measure for semantic distances by considering the context domain. In computer science, one of the biggest issues is to compare two objects which require a little bit of intelligence. It comes under cognitive science. Obviously it is a hard task until the two objects share some common attributes. This approach tends to compare semantic similarity by taking into account the target context from a given text corpus. For some cases, it will give very good results but it will depend on the target context and corpuses. In [9], Shi Bin et al, describes the semantic similarity by combining graph-based and information content based approaches. They have constructed the concept tree by using ontology and measured path length between the two concepts and integrated with information content (IC) and edge weights. IC is proportional to the information shared by the two concepts. More the information shared means closer. Based on this they measured closeness by finding the common ancestor concept of two concepts and calculated the distances. They forgot the descendents or hypernyms, the distance from this common descendent to the concepts might be smaller than the distance from the common ancestor. Our method considered both common ancestor and common descendent and took one that gave minimum distance between the two concepts.

In [10], Wanlong and Dayou, semantic similarity computation between concepts is proposed. They considered relation, property and instances of concepts and calculated similarity. They have used static weights for relation, property and instances. The level of granularity and density that have been considered are not specified. In [11], Wenjie Li and Qiuxiang Xia presented closeness between the concepts based on the analysis of distance and traditional methods. They have taken the ratio of semantic content and differences in depths between the nodes. They have omitted the concept of hypernyms and related concepts of hyponyms. Our approach considers the concepts available above and below the two concepts in the ontology. In [12 13], the relatedness has been computed based on wordNet structures. They have used hierarchical concept tree (HCT) and hierarchical concept graphs (HCG). These structures are built based on the hierarchical relations and considered ancestor nodes of related concepts of the given concept. Each structure has an ancestor and more descendent concepts which are all related and relevant at particular sense. It means that the node has more than one sense. The given concept may be close to one or more of these senses. On the other way, a word has more descendents means it is more independent. So, it is very difficult to say that they are close to each other. The main limitation of this is that it fails to conclude or find the descendents of two concepts.

3. Our Work

A framework has been provided for describing semantic similarity between two entities. It focuses mainly on two issues: one is extraction of related words using ontology WordNet, and the other issue is formalization of words in conceptual notation [14]. These can be described independently for the domain of semantic similarity. When applied to a specific problem of this type, the efficiency is extremely dependent on the extraction of glossary, threshold and granularity level of representation of the extracted thesaurus. Much more extensive words will be required for truly versatile and human acceptable similarity in conceptual form. So, it has to arrange in conceptual form in such a way that calculates true similarities. One of the ways of representation is Direct Acyclic Graph [15] (DAG).

DAG consists of both nodes and edges without any cycles. Here nodes are extensive words and edges represents the relations has-a and include. The nodes at edges are either hyponym or hypernym. The type of relation possessed between any two nodes is purely depending on the relation type. Based on these, there are two kinds of DAG graphs, specialization DAG (SDAG) and generalization DAG (GDAG). The DAG can be constructed in one of two approaches. SDAG follows top-down approach. GDAG follows bottom-up approach. Specialization DAG is constructed with only hyponyms of the given word. The given or input word acts as origin of the graph. All of its hyponyms are adjacent to that word in top-down manner i.e. hyponyms are siblings of that word. Now each sibling has its own hyponyms which are again add as siblings. It grows in this fashion until a specific threshold value is satisfied.

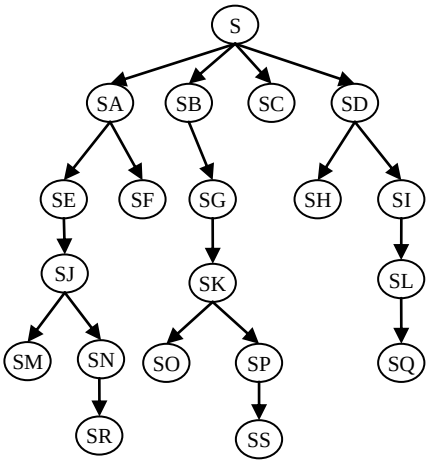

Fig. 3.1 Specialization DAG on concept S

The specialization DAG can be constructed using hyponyms of the terms and we called these terms as hypo nodes and edges are called as hypo arcs. A graph that consists of hypo nodes and hypo arcs is called specialization DAG. The fig. 3.1 is an example of SDAG. Specialization DAG is constructed with the help of top-down strategy. One of the two words is taken as an origin and is built as graph that has been constructed as shown in fig. 3.1 above. The starting node extracts hyponyms [1] just one level of its granularity from the wordNet[1] and arrange in an order below of that word and connect with direction called direct edge. The association between the two words connected by an edge is specialization. Repeat this until it reaches a level which matches to a specific threshold value. Extraction of related words is the result of an extraction algorithm. This algorithm will take a word as input and generate hyponyms for SDAG and produce that graph as an output. Generalization DAG is constructed with only hypernyms of the given word. The given input word acts as origin of the graph which starts at bottom. All of its hypernyms are adjacent to the original word in bottom-up manner i.e. hypernyms [1] are generalization concepts. Now each hypernym has its own hypernyms in the next higher level and these are again more generalization concepts than the previous concepts. Like this it grows till a specific threshold value is met. The relation between any two nodes in this graph also represents relation of hyponym. Both the graphs represent the same relation but the words might be different because of the approach of extraction of concepts and construction of graph is different. The generalization DAG graph can be constructed using hypernyms of the terms and we call these terms as hyper nodes or holonodes and edges are called as hyper arcs or holoarchs. A graph that consists of hyper nodes and hyper edges is called generalization DAG. For example, the following fig. 3.2 describes the Generalization DAG on concept S.

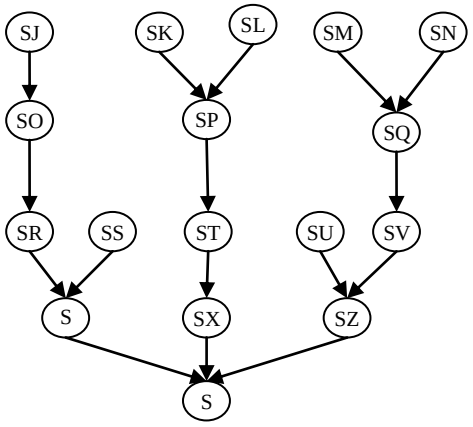

Fig. 3.2 Generalized DAG on S

3.1 Algorithm for specialization DAG

The extraction procedure begins with an empty graph as same as breadth-first-search and proceeds as described in the following.

Step 1. The original term (t) is added to the graph G as a node. G = {t}
Step 2. Initialize open with t and close with empty, i.e. open = {t} close = { } and $sets_0$ = {t} the index zero describe the level of the terms that are grouped at level zero. The number of terms in $sets_0$ is 1 i.e., length (L) of the $sets_0$ here is one.
Step 3. While the set open is not empty or not met its termination condition, do the following

i. Remove the leftmost term from the data structure open, add to the set close and generate its hyponyms (H) up to a specific threshold value is satisfied.
ii. $sets_i = \bigcup_{t_k \in sets_{i-1}} H_k$ Union of hyponyms of each term in $sets_{i-1}$. Here i indicate the level of the DAG.
iii. For each generated hyponym (t_k)
iv. Check the term t_k is matched with the second original term, if it is matched then return a flag SUCCESS.
v. Otherwise discard the term t_k if it is in open or close otherwise add remaining terms to the rightmost of the set open.
vi. End of the internal conditional control loop.

Step 4. End of the outer conditional control loop main.

3.2 Algorithm for generalization DAG

The extraction procedure begins with an empty graph as same as breadth-first-search and proceeds as described in the following.

Step 1. The original term (t) is added to the graph G as a node. G = {t}
Step 2. Initialize open with t and close with empty, i.e. open = {t} close = { } and $sets_0$ = {t} the index zero describe the level of the terms that are grouped at level zero. The number of terms in $sets_0$ is 1 i.e., length (L) of the $sets_0$ here is one.
Step 3. While the set open is not empty or not met its termination condition, do the following

i. Remove the leftmost term from the data structure open, add to the set close and generate its hypernyms (H) until a specific threshold value is satisfied.
ii. $sets_i = \bigcup_{t_k \in sets_{i-1}} H_k$ union of hypenyms of each term in $sets_{i-1}$. Here i indicates the level of the DAG.
iii. For each generated hypernym (t_k)
iv. Check the term t_k is matched with the second original term, if it is matched then return a flag SUCCESS.
v. Otherwise discard the term t_k if it is in open or close otherwise add remaining terms to the rightmost of the set open.
vi. End of the internal conditional control loop.

Step 4. End of the outer conditional control loop main.

The above algorithms are applied on a term (t) for generating two different DAG graphs. First DAG graph is built by first algorithm using only hypernyms of the term (t) called generalization DAG ($GDAG_t$). The second algorithm can build another DAG using hyponyms of the term (t) called specialization DAG ($SDAG_t$). In this way, we can generate or build four DAG for a pair of terms for which to calculate closeness between of them. For example, take two terms s and t as a pair to find the semantic closeness, generate four graphs: $SDAG_s$, $GDAG_s$, $SDAG_t$, and $GDAG_t$. From these graphs we can determine the two concepts Least Common General Concept (LCGC) and Least Common Specific Concept (LCSC). These two terms play key role in determination of semantic similarity of two terms.

The semantic similarity of two terms depends on their shared information [16]. These shared terms represent the closeness in concern graphs, i.e. LCGC the closest and nearest node subsumes of two distinct graphs $GDAG_s$ and $GDAG_t$. It bears the qualitative information content of shared information of these words. In case of Least Common Specific Concept (LCSC) which is least and closest common term that subsumes two distinct graphs $SDAG_s$ and $SDAG_t$. The least common specific concept of two terms s and t is the nearest and common child concept of both s and t. The details of these will be given in the next section since these will be determined and used in the calculation only after converting above graphs into undirected fragmentations of the concern concepts. In the next section a frame work that begins with the conversion of DAG into fragmented clusters on ontology for automatically discovering cluster from directed acyclic graphs of the original terms is given. The main object of this framework is to enable the extraction of concepts and to structure these into labeled clusters.

3.3 Fragmentations on Ontology

A graph G consists of two sets nodes or concepts V and relations or edges E. The set V is a finite, nonempty set of terms. The set E is a set of pairs of nodes called logical relations, in general as edges. In the above section we have constructed two DAG in two different ways. Each one carries a certain concept related to the beginning term. Here we call it as a concept. A concept describes thing semantically that relates to words which are closely related to that concept. The similarity relation from one term to another is same as in vice versa. We can describe the same in another way: the similarity from one word to second is same as the second word to first word. Here we have taken a relation as has-a, some applications might be taken as distance or closeness. In real world the distance or similarity is not in one way, it should be in both directions.

In previous section we have built four different graphs for a pair of terms. Convert all these graphs into undirected graphs and call it as fragments. For four directed acyclic graph, we get four fragments F_1, F_2, F_3, and F_4 with same nodes and edges. Fragmentation is the base for similarity. Similarity computation is derived from sets. The sets are created dynamically for each term and independent of each corpus. So, the semantic similarity of terms should be independent. This measure overcomes the limitation [17]. The limits of the semantic coverage of the child nodes are the partition of the semantic coverage of their parent nodes. That is, the terms subsumed by sibling terms are usually non-overlapping so, the relationship between two siblings is captured only through their ancestor or descendent node. Determination of common node is not done in one side. It should be looking in all directions in all possible combinations of F_1, F_2, F_3 and F_4.

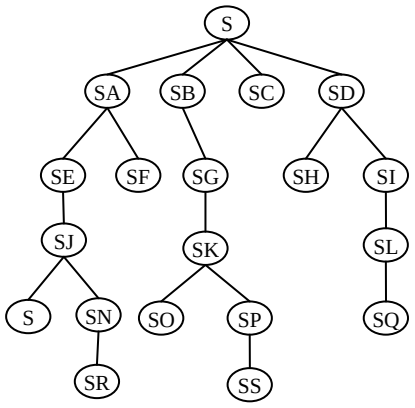

Fig. 3.3 Fragmentation of S with specialization on ontology

The above fig. 3.3 describes the fragmentation of concept S on ontology derived from the specialization DAG of S and is called as F_1. Similarly fig. 3.4 has derived from the generalization DAG of concept S called fragmentation of S with generalization on ontology. Here we call it as F_2. For the second concept T, derive two more fragments F_3 and F_4 from generalization DAG and Specialization DAG respectively.

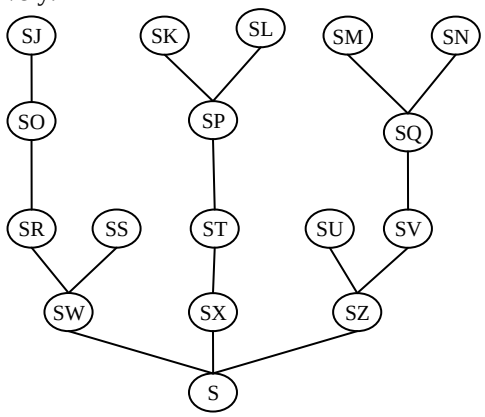

Fig. 3.4 Fragmentation of S with generalization on Ontology

The most important objective of the ontology-based fragmentation is semantics. The semantics can be measured by Information Content [18] of common term that can be determined from combinations of fragments. Semantic measures are calculated in two steps (i) term counting and (ii) determination of closest common term. Four different fragments describe semantics in four ways since the original terms position is not known in advance and it is determined dynamically by the system. For each pair of the terms, we get four different fragments with respect to specialization and generalization. Term count can be measured by counting number of term in that fragment i.e., size of the set of term in that fragment. Closest common term is measured by shortest path lengths between the original terms to common term. All these features are used to calculate the closeness of the concepts.

3.4 Semantic Similarity

Concepts can be described by words, but there are many different ways of doing it. One way is to be compared different methods and see how similar they are and take into the account that words are context dependent and related. Therefore their choice and combination has influence in the result. Determination of semantic similarity when there is no direct overlap [3] in the exact concept needs semantics, context and taxonomy with other corpus. Compare concepts from a similarity relational point of view to establish a semantic similarity relation. Using this idea on the other hand that words can be understood as imprecise concepts, the concepts that gradually related to other concepts are very tough to get its relations with other corpus. Extraction algorithm

developed to manage the corpus, interface to analyze the relations between concepts and concepts of subsumes. This allows us to take into an account the context and the point of view in order to do a proper analysis. The extraction algorithm has two main objectives, first to help us to analyze the idea expressed by words and sentences in a conceptual and relational way. Second, use it in the further to correlate user queries expressed by words as fragment and make it as possible clusters to identify the common attributes, measure distances and calculate closeness of given pair in all directions. Consider maximum closeness in all four combinations.

Set Based Similarity: The goal of ontology based similarity is to find the similarity between entities expressed in the form of fragmentations using ontology. Very often, these fragments are set based that are constructed for the purpose of measuring similarity between the entities. When individual representations are available, there is a very good opportunity for finding semantic similarities. When two fragments share the same set of individuals, similarity is highly facilitated. For example, if two fragments share exactly the same set of individuals, then there can be a strong presumption that these concepts represent a correct relation. The easiest way to compare concepts when they share some taxonomies is to test the intersection of their concept sets A and B and to consider that these concepts are very similar when $A \cap B = A = B$, $A \cap B = A$ or $A \cap B = B$ in more general. Sheth described in [19] how relationships and similarities are integrated primarily on the set relationships with equal, contain, contained-in, disjoint and overlap.

$$DiceSim(A,B) = \frac{2|A \cap B|}{|A| + |B|} \quad (1)$$

Where A, B are the sets of concepts of fragments represents the concept A and B on ontology. The problem is the ability to handle faults: small amount of incorrect data may lead the system to draw a wrong conclusion on domain relationships. Moreover, the dissimilarity has to be one when none of these cases apply. The chance of getting such cases are very less but in some cases such as by taking small value of threshold or thesauruses of the lexical concepts. This version of the symmetric difference is normalized by introducing constants. Using this semantic similarity on sets is more robust than equality. It is also possible to compare a similarity based on the length of the path between the original concepts via common concept. This will be the more prominent than the set based similarity.

Path-based similarity: The path based approach is a more natural and direct way of evaluating semantic similarity in a taxonomy. It estimates the distance i.e. length of the path from one node to the other, between the nodes which correspond to the concepts being compared. For a given multidimensional concept space, the conceptual distance can conveniently be measured by the geometric distance between the nodes representing the concepts. The distance should satisfy the properties of concepts that are in hierarchical taxonomy: zero property, symmetric property and positive property. The distance between two concepts is very simple and easy in a network that is built by using is-a property [20]. In more realistic scenario, the distances between any two nodes that are adjacent are not necessarily equal. Especially in weighted semantic networks the adjacent nodes distance is not necessarily equal. To determine the edge weight automatically, certain aspects should be considered in the implementation. In this work the weights are automatically determined based on the value of threshold constant (α). Most of these are typically related to the structural characteristics of a hierarchical network. Some imaginable features are: density of the fragmentation, height of the fragment, weights of edges and type links. In semantic similarity based on the path, there are four cases to be considered to find the common concept occurrence in fragments of two concepts on ontology. Section 3.1 describes fragments on ontology. For pair of two concepts, we can get four fragments. We can combine these fragments in into four clusters. Each combination gives a new fragment on ontology called cluster. These four clusters are: case 1) combine fragments of two generalization concepts, one on S and another on T on ontology, case 2) combine fragments of two specialization concepts, one on S and another on T on ontology, case 3) combine fragments of S with Generalization on ontology and T with specialization on ontology, case 4) combine fragments of S with specialization on ontology and T with generalization on ontology. Cases 3 and 4 come under hybrid fragments of generalization on S and specialization on T and vice versa. These four becomes four cases and calculate similarity of each case and take highest semantic similarity as closeness of our value. The fig. 3.5 describes combination of fragments $GDAF_s$ and $GDAG_t$ as a cluster with P as LCGC of both fragments.

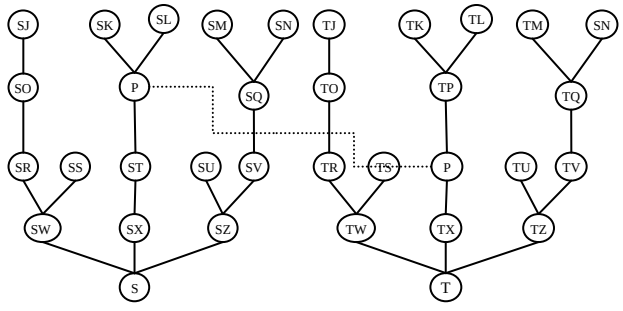

Fig. 3.5 LCGC P in Two Generalization Concepts S and T

Case 1 Similarity within the Generalization Fragment: To combine fragmentation of one concept (S) on generalization and concept (T) on generalization produce a fragment shown in fig. 1.11 with common concept P called Least Common General Concept (LCGC). Based on the principle, the shorter the distance between two concept nodes in the hierarchy, the more they are similar. Find the paths from P to concept S and the path from P to concept T. By using the following formula calculate the semantic similarity between the concept S and T using path based approach.

$$Sim_1(S,T) = 1 - \frac{\alpha * [(path(LCGC,S)+1) + (path(LCGC,T)+1)]}{2} + k \quad (2)$$

Case 2 Similarity within the Specialization Fragment: Join two fragmentations of concept S and T with specialization on ontology based on the Least Common Special Concept (LCSC) to get an undirected acyclic graph called tree. If LCSC is empty then the value of the constant is equal to $1+k=0$ and similarity is zero. One property of specialization of the concept is that the lower level pairs of concept nodes are semantically similar or closer than the node on higher level. Find the path from LCSC to concept S, the path from LCSC to the concept T and calculate similarity using the above formula.

$$Sim_2(S,T) = 1 - \frac{\alpha * [(path(LCSC,S)+1) + (path(LCSC,T)+1)]}{2} + k \quad (3)$$

Case 3 Similarity within the first Hybrid Fragmentation: In this case, the fragments of two concepts S and T that are constructed with generalization and specialization on ontology respectively joined by a common concept HP called hybrid fragment of specialization S and Generalization T. The lengths from S to HP and HP to T are used in the following formula to find the semantic similarity between two concepts.

$$Sim_3(S,T) = 1 - \frac{\alpha * [(path(HP,S)+1) + (path(HP,T)+1)]}{2} + k \quad (4)$$

Case 4 Similarity within the second Hybrid Fragmentation: This case is quite vice versa of case 3. Here to get a hybrid fragmentation, join the fragmentations of concept S with specialization and T with Generalization on ontology based on the common concept HQ and calculate semantic similarity by using the following formula. If there is no such concept, the semantic similarity is zero as same as in case 1.

$$Sim_4(S,T) = 1 - \frac{\alpha * [(path(HQ,S)+1) + (path(HQ,T)+1)]}{2} + k \quad (5)$$

Based on the foregoing knowledge, the semantic similarity based on the path approach between the concepts S and T can be computed by utilizing weighted summarized values of four cases. Here α and k are separately the weights of the relation on taxonomy structure of the ontology. These are required to meet restriction condition $\alpha + k = 1$.

$$Sim(S,T) = \max\{Sim_1, Sim_2, Sim_3, Sim_4\} \quad (6)$$

4. Experimental Results

The performance evaluation of measurement of semantic similarity between two concepts or words by the machine would be reasonable by comparing with the human commonsense on the same words. For evaluation, two important and well-known sets of concepts pairs are taken from data rated by experts from semantic similarity for general English. The first set collected by RG [Rubenstein and Goodenough, 1965] containing sixty five pairs of words covers fifty one subjects on sale from more similarity to more dissimilarity. The other data set was collected by MC [Miller and Charles, 1991] contains thirty pairs extracted from the sixty five pairs of RG, and covers nearly thirty eight subjects. This section uses both the data sets and compares results with results of both RG and MC data sets for testing. It would be reasonable to construct two fragments with all noun taxonomies in worNet with maximum threshold level value eight, since the semantically related concepts are always close and related to the origin concepts.

Table 4.1 Semantic Similarity between two words/concepts

SNo	Word Pairs	Sim_{set1}	Sim_{set2}	Sim_{frag1}	Sim_{frag2}	MaxSim
1	car journey	0	0	0.1	0	0.1
2	car travel	0.0181	0.00921	0.11111	0	0.1111111
3	glass necklace	0	0	0.55555	0.555555	0.5555555
4	glass mirror	0.08163	0.08333	0.9	0.9	0.9
5	bird eagle	0.02631	0.05194	0.83333	0.833333	0.8333333
6	cock bird	0.17475	0.20512	0.90909	0.909090	0.9090909
7	furnace stove	0	0.09302	0.5	0.5	0.5
8	magician wizard	0	0.45283	1	1	1
9	hill mound	0.27451	0.37209	1	1	1
10	autograph signature	0.37209	0.19607	0.875	0.875	0.875
11	forest shore	0.19230	0	0.6	1	1
12	forest woodland	0	0.34615	1	1	1
13	tool rooster	0.33333	0.05194	0.42857	0.5	0.5
14	tool implement	0.10958	0	0.875	1	1

15	pillow cushion	0.04301	0.32	0.875	0.875	0.875
16	lad grin	0.25	0	0.14285	1	1
17	gem jewel	0	0.47761	1	1	1
18	car ship	0.4375	0.02298	0.72727	0	0.727272
19	truck hovercraft	0.02298	0	0.72727	0.727272	0.727272
20	teacher lecturer	0	0.21621	0.875	0.875	0.875

The experimental results conforms both the information contents proposed by Miller Charles and Rubenstein and Goodenough provides a significant improvement over the traditional nodes or edge counting method. It also shows outperforms the information content approach. One should recognize that even a little percentage improvement over the existing approaches is of significance since the system is nearing the observed upper bound. The results are compared with the few more latest semantic similarity results [Resnik., 1995], [Shen Wan and Rafal, 2007] and [Jesus et al., 2011]. The comparison result is shown below.

Table 4.2 Semantic Similarity comparisons

SNo	Word Pair	Simfrag	SimResnik/SyMSS	Difference
1	car journey	0.1	0 (Resnik)	0.1
2	car travel	0.1111111	new combination	--
3	glass necklace	0.5555555	new combination	--
4	glass mirror	0.9	0.9 (resnik)	0
5	bird eagle	0.83333333	new combination	--
6	cock bird	0.9090909	0.489 (CV)	0.4200909
7	furnace stove	0.5	0.24 (SyMSS)	0.26
8	magician wizard	1	0.9999 (Resnik)	0.001
9	hill mound	1	0.39 (SyMSS)	0.61
10	autograph signature	0.875	0.33 (SyMSS)	0.545
11	forest shore	1	new combination	--
12	forest woodland	1	0.50 (SyMSS)	0.5
13	tool rooster	0.5	new combination	--
14	tool implement	1	0.9852 (resnik) 0.64 (SyMSS)	0.0148 0.36
15	pillow cushion	0.875	0.39 (SyMSS)	0.485
16	lad grin	1	new combination	--
17	gem jewel	1	1 (Resnik) 0.36 (SyMSS)	0 0.64
18	car ship	0.727272	new combination	--
19	truck hovercraft	0.727272	new combination	--
20	teacher lecturer	0.875	new combination	--

4. Conclusions

We have presented a method of measuring semantic similarity between two concepts or words. This utilizes concept elicitation in a hierarchical way as mean to determine the closeness of given two concepts. The experiments on single ontology and multiple clusters show the efficiency of the proposed approach. Since the evaluation of the closeness shows the improvements in accuracy that was achieved over existing traditional and other semantic similarity methods. The further work in this is to include granularity on fragmentations when building the concerned clusters and experiment the measures on ontology. With this, we can extend to analyze the short sentences or phrases and test the method with different phrases. Other possibilities of further work are related with the applications of the NLP tasks.

References

[1] G.A.Miller, "WordNet: A Lexical Database for English", Comm. ACM, Vol. 38, No.11, pp. 39-41, 1995.
[2] R.Richardson and A.FSmeaton, "Using wordNet in a Knowledge-based Approach to Information Retrieval", Working Paper CA-0395, School of Computer Applications, Dublin City University, Dublin, 1995.
[3] C.Buckley, G.Salton, and J. Allan, "The Smart Information Retrieval Project", In HLT'93 proc. of the workshop on

Human Language Technology, Morristown, NJ, USA, 1993. Association for Computational Linguistics. pp 392- 392.

[4] S.Banerjee and T.Pedersen, "Extended Gloss overlaps as a measure of Semantic Relatedness", In Proceedings of IJCA, Mexico, pp. 805-810, August 2003.

[5] Ce Zhang, Yu-Jing Wang, Bin Cui, and Gao Cong, "Semantic Similarity Based on Compact Ontology", ACM proc., of 17th International Conference on WWW, pp 1125-1126, 2008.

[6] Chen, Horng and Lee, "Document Retrieval Using Fuzzy Valued Concept networks", IEEE Transactions on Systems, Man and Cybernetics, Vol. 31, pp. 111-118, 2001.

[7] Hisham AI-Mubaid and Hoa A.Nguyen, "A Cross-cluster Approach for Measuring Semantic Similarity between Concepts", IEEE proceedings of International Conference on Information Resource and Integration, pp. 551-556, 2006.

[8] Ahmad EI Sayed, Hakim Hacid and Djamel Zighed, "A New Context-Aware Measure for Semantic Distance Using a Taxonomy and a Text Corpus" In Proceedings of the IEEE International conference on Information Reuse and Integration, pp. 279-284, August 2007.

[9] Shi Bin, Fang Liying, Yan Jianzhuo, Wang Pu and Zhao Zhongcheng, "Ontology-based Measure of Semantic Similarity Between Concepts", In proceedings of World Congress on Software Engineering, pp. 109-112, 2009.

[10] Wanlong Li, Dayou Liu, Shanhong Zheng and Suyun Jia, "A Navel Computational Approach to Concept Semantic Similarity", In proceedings of IC on Computer, Mechatronics, Control and Electronic Engineering, pp. 89-92, 2010.

[11] Wenjie Li and Qiuxiang Xia, "AMethod of Concept Similarity Computation Based on Semantic Distance", Prodedia Engineering, vol. 15, pp. 3854-3859, 2011.

[12] Hongzhe Liu, Hong Bao and De Xu, "Concept Vector for Semantic Similarity and Relatedness based on WordNet", The Journal of Systems and Software, vol. 85, pp. 370-381, 2012.

[13] Jesus Oliva, J.I.Serrano, M.D.del Castillo and A.Iglesias, "SyMSS: A Syntax-based Measure for Short-text Semantic Similarity", vol. 70, pp. 390-405, 2011.

[14] T.Tedersen, S.Patwardhan, and J.Michelizzi, "WordNet::Similarity-Measuring the Relatedness of Concepts", In the Proceedings of the 19th National conference on Artificial Intelligence, pp. 1024-1025, San Jose, CA, July'2004.

[15] Q.Peng, L.Zhao, Y.Yu and W.Frang, "A New Measure of Word Semantic Similarity based on WordNet Hierarchy and DAG theory", In Proceeding of International Conference on Web Information Systems and Mining", pp. 181-185, 2009.

[16] N Seco, T Veale, and J Hayes. An Intrinsic Information content Metric for Semantic similarity in WordNet . In Procedings of ECAI 2004, the 16th European Conference on Artificial Intelligence, Valencia Spain, 2004.

[17] Hongzhe Liu, Hong Bau, and De Xu. Concept Vector for Semantic similarity and Relatedness Based on WordNet Structure. Journal of Systems and software, vol. 85, pp 370-381, 2012.

[18] Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy, Proceedings of the 14th International Joint Conference on Artificial Intelligence, Vol. 1, pp 448-453, Montreal, August 1995.

[19] Amith Sheth, James Larson, Sloysius Cornelio, and Shamkant Navathe. A tool for integrating conceptual schemas and user views. In Proc. 4th international Conference on Data Engineering (ICDE), pp 176-183, Los Angeles, CA US, 1988.

[20] R Rada, H Mili, E Bichnell, and M Blettner. Delevlopment and Application of a Metric on Semantic Nets, IEEE Trans. Systems, Man and Cybernetics, Vol. 9, No. 1, pp. 17-30, jan. 1989.

First Author Mr. B.Bhaskara Rao is presently working as Associate professor in the Department of Information Technology, GIT, GITAM University, Visakhapatnam. He presented several research papers in national and International conferences and seminars. He published a good number of papers in national and International journals. He guided several students for getting their M.Tech degrees in Information Technology. His current research interests are Semantics Similarity, Information Retrieving, Internet Technologies and Data Mining.

Second Author Dr. V. valli Kumari is presently working as Professor in the Department of Computer Science and Systems Engineering, Andhra University and also a Director, Andhra University Computer Centre, Andhra University, Visakhapatnam. She holds Ph.D degree from Andhra University. She is awarded the best researcher award - 2014 in January, 2014. She obtained gold Medal for best research for her thesis, in the 75th Diamond Jubilee Convocation in Andhra University-2008. She published several research papers and delivered invited lectures at various conferences, seminars and workshops. She got best paper award in conference (INDICON-2011). She guided a number of students for their Ph.D and M.Tech degrees in Computer Science and Engineering and Information Technology. She is a fellow of IETE and founder vice chair for IEEE Vizag Bay Sub-section. Her current research interests are Privacy issues in Data Mining and Web Technologies, Image Processing, Network security and cryptography, Data Mining, E-commerce, Agent Software etc.