Computación y Sistemas
ISSN: 1405-5546
computacion-y-sistemas@cic.ipn.mx
Instituto Politécnico Nacional
México

Majumder, Goutam; Pakray, Partha; Gelbukh, Alexander; Pinto, David
Semantic Textual Similarity Methods, Tools, and Applications: A Survey
Computación y Sistemas, vol. 20, núm. 4, 2016, pp. 647-665
Instituto Politécnico Nacional
Distrito Federal, México

ISSN 2007-9737

Semantic Textual Similarity Methods, Tools, and Applications: A Survey

Goutam Majumder[1], Partha Pakray[1], Alexander Gelbukh[2], David Pinto[3]

[1] National Institute of Technology Mizoram, Aizawl, India

[2] Instituto Politécnico Nacional, CIC, Mexico City, Mexico

[3] Benemérita Universidad Autónoma de Puebla, Faculty of Computer Science, Mexico

{goutam.nita, parthapakray, davideduardopinto}@gmail.com, gelbukh@gelbukh.com

Abstract. Measuring Semantic Textual Similarity (STS), between words/ terms, sentences, paragraph and document plays an important role in computer science and computational linguistic. It also has many applications over several fields such as Biomedical Informatics and Geoinformation. In this paper, we present a survey on different methods of textual similarity and we also reported about the availability of different software and tools those are useful for STS. In natural language processing (NLP), STS is a important component for many tasks such as document summarization, word sense disambiguation, short answer grading, information retrieval and extraction. We split out the measures for semantic similarity into three broad categories such as (i) Topological/Knowledge-based (ii) Statistical/ Corpus Based (iii) String based. More emphasis is given to the methods related to the WordNet taxonomy. Because topological methods, plays an important role to understand intended meaning of an ambiguous word, which is very difficult to process computationally. We also propose a new method for measuring semantic similarity between sentences. This proposed method, uses the advantages of taxonomy methods and merge these information to a language model. It considers the WordNet synsets for lexical relationships between nodes/words and a uni-gram language model is implemented over a large corpus to assign the information content value between the two nodes of different classes.

Keywords. WordNet taxonomy, natural language processing, semantic textual similarity, information content, random walk, statistical similarity, cosine similarity, term-based similarity, character-based similarity, n-gram, Jaccard similarity, WordNet similarity.

1 Introduction

In Natural Language Processing (NLP), semantic similarity plays an important role and one of the fundamental tasks for many NLP applications and its related areas. Semantic Textual Similarity (STS) can be defined by a metric over a set of documents with the idea is to finding the semantic similarity between them. Similarity between the documents is based on the direct and indirect relationships among them [13, 49]. These relationships can be measured and recognized by the presence of semantic relations among them. Identification of STS in short texts was proposed in 2006 in the works reported in [30, 35]. After that, focus was shifted on large documents or individual words.

After that, since 2012 the task of semantic similarity is not only limited to find out the similarity between two texts, but also to generate a similarity score from 0 to 5 by different SemEval tasks[1]

[1] http://ixa2.si.ehu.es/stswiki/index.php/Main_Page

[1, 2, 3]. In this task, a scale of 0 means unrelated and 5 means complete semantically equivalence.

Since its inception, the problem has seen a large number of solutions in a relatively small amount of time. The central idea behind the most solution is that, the identification and alignment of semantically similar or related words across the two sentences and the aggregation of these similarities to generate an overall similarity [23, 35, 65].

One of the major goals of STS task, is to create a unified framework by combining several independent semantic components in order to find their impact over several NLP tasks. Developing such framework is an important research problem, having many important applications in NLP such as information retrieval (IR) and in digital education like text summarization [4, 61], question answering [37], relevance feedback and text classification [47], word sense disambiguation [30], and extractive summarization [50].

Semantic similarity also contributes for many semantic web applications like community extraction, ontology generation and entity disambiguation. It is also useful for Twitter search [50], where it is required the ability to accurately measure semantic relatedness between concepts or entities. In IR one of the main problems is to retrieve a set of documents and retrieving images by captions [11], which is semantically related to a given user query in a web search engine.

In the database field also, text similarity can be used for schema matching to solve the semantic heterogeneity for data sharing system, data integration system, message passing system, and peer-to-peer data management system [19]. It is also useful for relational join operations in database where join attributes are textually similar to each other. It has a verity of application domain including integration and querying of data from heterogeneous resources, cleaning of data, and mining of data [12, 51].

In NLP it is noticed that, STS is related to both textual entailment (TE) and paraphrasing, but differs in a number of ways. In NLP, TE can draw three directional relationships between two text fragments while the task considered two text fragments as text (t) and hypothesis (h) respectively. On the other hand, paraphrasing identification is the task of recognizing text fragments with approximately the same meaning within a specific context. So, TE and paraphrasing gives a yes/no decision and STS identifies the degree of equivalence of text and rated them on the basis of their semantic relationships.

Measuring semantic similarity between texts can be categorized into the following ways: (i) topological (ii) statistical similarity (iii) semantic based (iv) vector space model (v) word alignment based and (vi) machine learning. Among these methods, topological studies plays an important role to understand intended meaning of an ambiguous word, which is very difficult to process computationally. For many NLP related task it is important to understand the semantic relation between the word/ concepts. To decompose such systems we need to work with word level relation and those can be considered as hierarchical, associative and equivalence.

The rest of this paper is organised as follows: in Section 2, we reported the related work on topological methods. In Section 3, we examined the details implementation of three similarity methods. Topological methods are reported in Section 3.1 and statistical similarity measures are reported in Section 3.2. In Section 3.3, we reported string based similarity measures in two categories such as character based and term based similarity (Section 3.3.1 and 3.3.2). We also proposed a new method for detection of textual similarity between sentences based on language model and WordNet taxonomy in Section 4 and we reported a short experiment on the proposed method in Section 5. In Section 6, we listed out available software's and tools those are used for measuring the similarity. Applications of STS are reported in Section 7, and finally conclusion of the work is drawn in Section 8.

2 Related Work on Topological Methods

In many cases determining the intended meaning of an ambiguous word is difficult for human and it is quite difficult to process automatically also. This ambiguity can be eliminated by considering the following relationships among

the words or concepts: (i) hierarchical (e.g. IS-A or hypernym-hyponym, part-whole etc.), (ii) associative (e.g. cause-effect) and, (iii) equivalence [25]. Among these, IS-A relation is widely used and studied, which maps to the human cognitive view of classification (i.e. taxonomy). The IS-A relation among the concepts has been suggested and employed as a special case of semantic similarity of distance [42]. Semantic similarity can be estimated by defining a topological similarity by using ontologies to define the distance between term and concepts.

Taxonomy is often represented as a hierarchical structure and also considered as a network structure. To measuring the similarity information of this network can be useful. There are several ways to determine the conceptual similarity between two words in a hierarchical semantic network. There are essentially two types of approaches, which calculate topological similarity between ontological concepts. Those are (i) node based (information-content approach) and (ii) edge-based (distance based).

Issues related to lexical association was reported in [45], where a generalization technique of lexical association was proposed. To solve these issues (i.e. reliable word/ word correspondence) author facilitate different statistical facts by considering word classes rather than individual words. In this task, a set of possible word classes were constructed from WordNet [36] and an investigation was conducted to identify the relationship between word/ classes using mutual information. For word–based information retrieval, information from WordNet was passed over a SMART environment where a content description was added (only part-of-speech information) with the input text.

Ambiguity of word form during document indexing was investigated using a semantic based network where semantic distance between network nodes was considered [62]. In this work, word sense during document indexing was studied using the WordNet semantic network. Distance between multiple senses of input word was disambiguated by finding the combination of senses from a set of contiguous terms.

In another work, Philip Resnik proposed a method for identification of semantic similarity in a taxonomy based on the notion of information content [45]. Similarity between two words/ concepts was evaluated by considering the common information between them and a set of fifty thousand (50,000) nodes form WordNet taxonomy of noun class was considered for this task. To calculate the frequencies of concepts Brown Corpus of American English (having 1000,000 words) was considered [27].

Jiang and Conrath introduces a new approach for measuring semantic similarity between words using lexical taxonomy structure with corpus statistical information. So the semantic distance between nodes in the semantic space was constructed by the taxonomy, which provides a better result with the computational evidence those are derived from a distributional analysis of corpus data. This proposed method, is a combined approach in which edge counting scheme was inherited and further enhanced by node based approach [25].

To find the similarity between phrases and sentences a random walk over a graph was proposed in [43]. In this work, local semantic information and semantic resources of WordNet was combined together. Semantic signature generated by random walk was compared to another such distribution to get the similarity score. It is identified that, graph work similarity between texts can be used as feature for recognizing textual entailment task.

Methods reported in this section are based on topological similarity between ontological concepts and apart from these, methods related to ontological instances namely: (i) pair–wise; and (ii) group–wise are also found in literature. It was founded that methods based on ontological instances are mainly used to represent medical knowledge's and no such work was noticed, which was used for semantic textual similarity between classes or phrases or sentences. So all these tasks are not reported here, because the proposed work is planned for textual similarity only. In the next section a detailed illustration is reported for methods used to identify the semantic similarities

between words/ classes based on taxonomical concepts.

3 Semantic Similarity Methodologies

3.1 Topological/ Knowledge-based Methods

In the field of Information Retrieval (IR), document retrieval based on semantic similarity of words has been largely investigated and all these methods consider the semantic and ontological relationships that exist between the words (e.g. polysemy, synonym etc.). So based on this knowledge semantic similarity between objects in ontology can broadly be categorised into three groups like: (1) node-based; (2) edged-based; and (3) hybrid where it combines node and edge-based.

1. *Node Based/ Information Content Approach:* Node based or Information Content (IC) approaches [44], [45], are used to determine the semantic similarity between concepts. In this method, each of the concept or node poses IS-A taxonomy are kept in one set called C and all of these nodes carry unique concepts. Intuitively, one key to the similarity of two concepts is that to which they share information in common. In taxonomy direct relation between two concepts can be found by an edge counting method. In this method, if the minimal path between two nodes is long, that means it is necessary to go high in the hierarchy to find a least upper bound. An example of IS-A relationship between the concepts is shown in Fig. 1, where two concepts *NICKEL* and *DIME* both subsumes *COIN*. In this example *NICKEL* and *CREDIT CARD* shares a common super class *MEDIUM OF EXCHANGE* [64].

   In order to avoid unreliability edge-distances between nodes, it is possible to associate probability with taxonomy. The value of IC of a class is obtained by estimating the probability in a large corpus with a function $p : C \rightarrow [0, 1]$ if $c \in C$, $p(c)$, being the probability of encountering an instance c. Considering the notation of information theory [52], IC of a class can be calculated as follows:

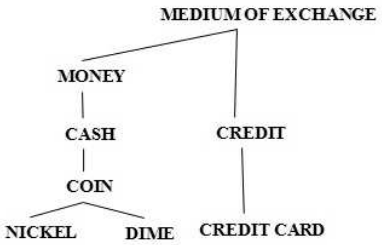

Fig. 1. Representation of a WordNet Taxonomy (IS-A Relationship) [64]

$$IC(c) = log^{-1}p(c). \quad (1)$$

Quantifying information content in this way: if the probability increases, its information content decreases. It means that if there is a unique top in the tree, then its probability is 1, so the information content is 0 and the similarity of two concepts can be calculated as follows:

$$sim\,(c_1, c_2) = max_{c \in S(c_1,c_2)}\,[-log\,p\,(c)]\,, \quad (2)$$

where $sim(c_1, c_2)$ is the set of concepts that subsume both c_1 and c_2. From Fig. 1, it is identified that, similarity of *NICKEL* and *DIME* can be calculated by considering all the upper bounds. Among those upper bounds node having highest information content value is considered as similarity score between these two nodes.

To implement the information content model reported in [61], WordNet fifty thousand nodes were considered, where taxonomy of concepts represented by nouns and compound nominals [36]. Before implementing IC, two concepts need to define as sets of $words(c)$ and $classes(w)$. $Words(c)$ is the set of words subsumed by the class and $classes(w)$ is defined as the classes in which the word is contained. The class can be seen as a sub-tree in the whole hierarchy and $classes(w)$ is the set of possible senses that the word has:

$$classes\,(w) = \{c|w \in words\,(c)\}\,. \quad (3)$$

A simple class/ concept frequency formula is also defined in [45] and [46], where the number of word sense is the key factor:

$$freq\,(c) = \sum_{w \in words(c)} freq\,(w) \qquad (4)$$

and

$$freq\,(c) = \sum_{w \in words(c)} \frac{freq\,(w)}{|classes\,(c)|}. \qquad (5)$$

Finally, the class probability can be computed using Maximum Likelihood Estimation (MLE):

$$p\,(c) = \frac{freq\,(c)}{N}, \qquad (6)$$

where N is the total number of nouns observed (excluding those not subsumed by any WordNet class, of course).

An example of multiple inheritance concepts like *NICKLE* and *GOLD* those have more super classes as shown in Fig. 2. In this case one word have more sense, so the similarity can be determined by the best similarity value among all the class pairs, which belongs to their various senses [25]:

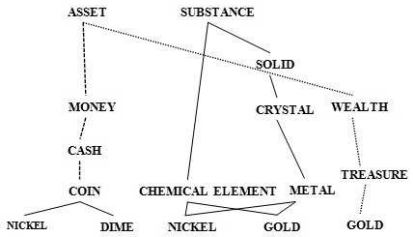

Fig. 2. WordNet Taxonomy of Multiple Inheritance [25]

$$sim\,(w_1, w_2) = max_{c_1 \in sen(w_1), c_2 \in sen(w_2)} [sim\,(c_1, c_2)], \qquad (7)$$

where $sen(w)$ denotes the set of possible senses for word w.

In another task [43], extends the node based method to vector space model for semantic measure using random walk algorithm. In this approach, instead of comparing between two text segments directly, it compares distribution of each text and a random walk is generated over a graph, which is derived from WordNet and corpus statistics.

WordNet is itself a graph over clusters, which contains one sense of one or more similar words. Each node in the graph represents a synset. Word having different meaning: multiple synsets (or cluster) is generated based on different meaning. For example the word *bank* belongs to the two different synsets, one for financial bank and other for river bank. By constructing each edge created from a WordNet relationship is guaranteed to have a corresponding edge in the opposite direction. Nodes are connected with edges (represents the relation) corresponding too many relationships from WordNet is as follows: hypernym/hyponym, instance/instanceof, all holonym/meronym links, antonym, entails/entailed by, adjective satellite, causes/caused by, participle, pertains to, derives/derived from, attribute/has attribute, and topical (but not regional or usage) domain links. Following types of nodes from WordNet was considered for graph construction:

(a) **Synset:** Each WordNet synset has a corresponding node. For example, one node corresponds to the synset referred to by "dog#n#3," the third sense of dog as noun, whose meaning is "an informal term for a man."

(b) **TokenPOS:** One node is allocated to every word coupled with a Part Of Speech (PoS), such as "dog#n" meaning dog as a noun. These nodes link to all the synsets they participate in, so that "dog#n" links the synset nodes for canine, hound, hot dog etc.

(c) **Token:** Nodes do not have any part-of-speech information in synsets, all the *TokenPOS* nodes were linked with all such nodes.

Random walk methods have following advantages over traditional node based method:

(a) It enables the similarity measure to have a principled means by combining multiple types of edges from WordNet.

(b) By traversing all the links, random walk aggregates the local similarity statistics across the entire graph [22].

A random walk of an undirected weighted graph was defined with transition probability between the links of the elements of database, which is designed with WordNet. So, a random walker can jump from element to element and each element of Markov–chain is represented as state into the taxonomy. Finally, the similarities between text passages was computed using Markov–chain Model [16].

In Markov-chain model, a weighted graph G with weight w_{ij} between node i and j was considered, where the database elements and links represents node and edges of the graph. The weight w_{ij} must have following convention: the relation between i and j is more, the larger the value of w_{ij} and the walk should be minimum and the value of $w_{ij} > 0$ and $w_{ij} = wji$.

Sequence of node visited by a random walker are called a random walk and described by a Markov–chain. A random variable $s(t)$ contains the current state of the Markov–chain at time t: if the random walker is in state i at time t, then $s(t) = i$. The random walk is defined with the following single - step transition probabilities of jumping from any state or node $i = s(t)$ to an adjacent node j as follows:

$$j = s\left(t+1\right) : P\left(s\left(t+1\right) = j | s\left(t\right) = i\right) = \frac{a_{ij}}{a_{i.}} = P_{ij}, \quad (8)$$

where $a_{i.} = \sum_{j=1}^{n} a_{ij}$ and a_{ij} is the elements of symmetric adjacency matrix A of the graph and defined as $a_{ij} = w_{ij}$, if i and j is connected else 0.

We need to compare the stationary distribution of two Markov–chains of two text passages to calculate the semantic relations between them. The transition probability $n_i^{(t)}$ of finding the particle of any node as the sum of all ways in which the particle could have reached n_i from any other node at the previous time step as follows:

$$n_i^{(t)} = \sum_{n_i \in V} {n_j}^{(t-1)} P\left(n_i | n_j\right), \quad (9)$$

where $P(n_i|n_j)$ is the probability of transitioning from n_j to n_i.

2. *Edge-based/ Distance Approach*: The edge based approach is the direct way of computing semantic similarity in taxonomy. It counts the number of edges between two nodes those corresponds to the concepts being compared. Minimum the path between two nodes they are more similar [25].

   It was pointed out that, in a hierarchical taxonomy distance between nodes must satisfy the matrix properties like: zero property, semantic property, positive property. Because of distance between two adjacency nodes should not necessarily equal, so it is necessary an edge connecting two nodes must be weighted. To determine the weight following structural characteristics should be considered [42]:

   (a) **Network Density:** higher densities in WordNet need to consider for example plant–flora section in WordNet for measuring the network density. Distance between the nodes is closer to the local density which is reported in [46].

   (b) **Node Depth:** in terms of the depth it can be said that, distance shrinks as one descends down a hierarchy.

   (c) **Type of link:** it represents the relation between two nodes. In many edge-based model only IS-A link is considered [21, 42]. Other relations can also consider such as Meronym-Holonym, which have different effect for calculating the weight.

(d) **Link strength of specific child link:** this could be measured using WordNet relationships between child node and its parent node.

Weight measurement also done manually for the edges and those works are reported in [18, 21, 42, 66]. To measure weight automatically, certain observations were considered over the Hierarchical Concept Graph (HCG). For measuring the weight of a link, density, depth of the HCG and link strength between child and parent nodes is considered [46]. The density of a HCG for a specific link type is estimated by counting the number of links of that type. The strength between the links was estimated as a function of nodes IC value and its sibling and parents node. Finally, results of these two operations were normalized by dividing the depth of the link.

A minimum and maximum range was taken before measuring the weight between two nodes [62]. Because of an edge represents two inverse relations, the final weight of an edge was fixed by averaging the two weight values. The depth–relative scaling process was adopted in which the average value is divided by the depth of the edge within the overall tree. The weight of an edge between two adjacent nodes n_1 and n_2 was calculated in the following way:

$$w\left(n_1, n_2\right) = \frac{w\left(n_1 \rightarrow_r n_2\right) + w\left(n_2 \rightarrow_{r'} n_1\right)}{2d}, \tag{10}$$

given

$$w\left(X \rightarrow_r Y\right) = max_r - \frac{max_r - min_r}{n_r\left(X\right)}, \tag{11}$$

where $\rightarrow_r$ is a relation of type r, and $\rightarrow_{r'}$ is its reverse, d is the depth of the deeper one of the two and $n_r\left(X\right)$ is the number of relations of type r leaving node X.

The value of 0 was assigned for all synonym type relations. Holonymy, hyponymy, hypernymy, and meronymy are the types of relation, where weights ranging from 1 to 2 and for antonym type relation weights was assigned as 2.5.

Edge counting method has been considered to determine the edge based similarity [45]. To convert the distance measure to the similarity measure, by subtracting the path length from the maximum possible path length as follows:

$$sim_{edge}\left(w_1, w_2\right) = 2d_{\max} - \left[\min\left(c_1, c_2\right) len\left(c_1, c_2\right)\right] \tag{12}$$

where $d_{\max}$ represents the maximum depth in the taxonomy, and then c_1 and c_2 ranges over senses of word w_1 and w_2 respectively.

3. *Hybrid Approach*: Node and edge based methods discussed in previous sections have many differences in between them. The edge-based methods, looks true without any concise reasoning and on the other hand, node-based approach looks more accurate than distance-based. The distance measure was relayed on the subjective knowledge of the network while the WordNet was used not for measuring the similarity, but for the construction of the network layers.

   On the other side information content was not sensitive to the link types [45], but it is dependent on the structure of the taxonomy. Although these two methods are different from each other, a combined method was derived from edge-based while it considers the information content as a decision factor [25]. In this method, a link strength factor was first considered by taking the conditional probability of the child concept c_i of its parent concepts p:

$$P\left(c_i|p\right) = \frac{c_i \cap p}{P\left(p\right)} = \frac{P\left(c_i\right)}{P\left(p\right)}. \tag{13}$$

   The link strength (LS) was defined by considering the negative logarithm of the conditional probability (see Eq.(14)), by following the argument of information theory (see Eq.(1)) as follows:

$$LS\left(c_i, p\right) = -log\left(P\left(c_i|p\right)\right) = IC\left(c_i\right) - IC\left(p\right). \tag{14}$$

   From Eq.(14) it is clearly understood that, the difference of information content values be-

tween child and parent has been considered as *LS*.

By considering other structural characteristics mentioned edge-based approach also considered here to calculate the weight wt of a child node as follows:

$$wt(c,p) = \left(\beta + (1-\beta)\frac{\bar{E}}{E(P)}\right)\left(\frac{d(p)+1}{d(P)}\right)^{\alpha} [IC(c) - IC(p)]\, T(c,p), \quad (15)$$

where $d(p)$, and $E(P)$ denotes the depth and the local density of the node p respectively and $\bar{E}$ represents the average density in the tree and $T(c,p)$ is the link type factor. The parameters α and β controls the degree of depth and density to calculate the edge weight. So the distance between two nodes is the summation of the edge weights and a shortest path between them:

$$Dist(w_1, w_2) = \sum_{c \in \{path(c_1,c_2) - LSuper(c_1,c_2)\}} wt(c, parent(c)). \quad (16)$$

3.2 Statistical/ Corpus-Based Similarity

Statistical similarity learned from data (i.e corpus), which is a collection of written or spoken text. In this method, a statistical model was build first and then similarity is estimated. Several models have been proposed during the past few years and we found following categorization:

1. *Latent Semantic Analysis (LSA)*: In this method, the contextual information of words have been extracted and represented from a large corpus of text [28]. In the first step, text is represented as a matrix in which rows and column represents the unique words and text segments. Each entry represents the frequency count of the word, which appears in the text [29]. Cell frequencies are weighted by a function, which expresses two meanings: (1) importance of a word in a text and (2) the degree to the word type sharing information in the domain of discourse. Two ways it can be implemented: (1) as a similarity matrix between the words and text segments for the contextual usage taken as practical expedient; and (2) as a computational process model to represent the underlying substantial portions of the acquisition and utilization of knowledge. To reduce the number of rows the Singular Value Decomposition (SVD), is used while preserving the similarity structure among the columns. To measure the similarity the cosine angle between the word vectors is considered, which is formed by any two rows.
2. *Generalized Latent Semantic Analysis (GLSA)*: Performance of LSA degrades when word vectors are generated from text corpus, which is heterogeneous in nature [5]. GLSA framework is used to find the terms and document vectors, based on semantically motivated pair–wise term similarities, instead of bag–of–words document vectors, which is used in LSA [48]. It is a framework in which different measures of semantic associations of terms are combined with different methods of dimensionality reduction [33]. Implementation of GLSA is as follows:
   (a) Consider that, D, V and C as a set of documents, vocabulary and large web corpus.
   (b) Constructed a weighted term–document matrix M based on D.
   (c) For the words in V, obtain a pair–wise similarity matrix S, based on the corpus C.
   (d) By preserving the similarities obtain a low dimensional vector space representation as $Z^T_{k\times V}$, where k is the dimension of the matrix.
   (e) Finally compute the document vectors by combining term vectors as
   $$\hat{M} = Z_k^T D, \quad (17)$$
   where columns of $\hat{M}$ represents the documents in the k–dimensional space.
3. *Explicit Semantic Analysis (ESA):* In this method, the meaning of any text is repre-

sented as a weighted vector of Wikipedia–based concepts. With the help of machine learning techniques, representation of vectors have been done over a high–dimensional space. This method is useful for fine–grained semantic representations of unrestricted natural language text [17]. To represent the text as a weighted mixture of natural concepts, the Wikipedia articles are used, because it is a collection of largest encyclopedia, which is defined by humans and can be easily explained. Semantic *interpretation vectors* were build to maps the natural language fragments to a weighted sequence of Wikipedia concepts based on the their order of the input. Semantic similarity is computed by comparing the vectors, using the cosine metric [68]. This semantic analysis is *explicit* in nature, because meaning of concepts is done on human cognition, rather *latent concepts* used in LSA.

4. *Pointwise Mutual Information – Information Retrieval (PMI – IR):* It is a unsupervised learning algorithm, to recognizing the synonym of a problem word from a set of alternative words. This algorithm uses any search engine to issuing a search query and analyze the query result to find the synonym word. The unsupervised learning algorithm uses the Pointwise Mutual Information (PMI), to analyze the statistics of data, which is collected by the search engine, i.e., Information Retrieval (IR). The performance of the method depends on two things: (1) power of the search engine query language and (2) indexing of the search engine (i.e., collection of documents).

5. *Normalized Google Distance (NGD)*: This method is feature free and uses the web and search engine to provide the contents and automatically generates the semantic relations between the words and phrases [10]. The drawback of this method is, it completely depends the accuracy of search engine and it was also noted that, google count can be inaccurate when search queries included the OR operator [6]. To find the similarity between two strings s_1 and s_2 it has the following steps:

   (a) First find the information distance and denoted as $E(s_1, s_2)$ as follows:

   $$E(s_1, s_2) = K(s_1, s_2) - min[K(s_1), K(s_2)], \tag{18}$$

   where $K(s_1, s_2)$ is the Kolmogorov complexity of a compressor, which produces the shortest binary length of the pair s_1, s_2.

   (b) The next step is to find a distance D by considering all *admissible distances*, whose length is equal to a prefix program of given s_1, s_2, has the equal binary length of the distance $D(s_1, s_2)$ and then

   $$E(s_1, s_2) \leq D(s_1, s_2) + c_D, \tag{19}$$

   where c_D is a constant and it can be said that $E(s_1, s_2)$ *minorizes* $D(s_1, s_2)$ up to an additive constant.

   (c) Third step is to find the *normalized information distance (NID)*, for every pair of string and generate a similarity score in between 0 and 1, on the basis of the feature in which they are similar. The NID is defined as

   $$NID(s_1, s_2) = \frac{K(s_1, s_2) - min[K(s_1), K(s_2)]}{max[K(s_1), K(s_2)]}. \tag{20}$$

   (d) Next we need to measure *Normalized Compressed Distance (NCD)*, because NID is limited, where $K(s_1)$ is the number of shortest code and s_1 can be decompressed. So the NCD is

   $$NCD(s_1, s_2) = \frac{C(s_1 s_2) - min(C(s_1), C(s_2))}{max[C(s_1), C(s_2)]}, \tag{21}$$

   where C denotes the compressor and $C(s_1)$ denotes the length of the compressed version on s_1.

   (e) Now, to find the google similarity distance between s_1 and s_2, the Google code of length $G(x)$ is considered. It represents the shortest expected prefix code of the

associated Google event s_1. Next Google distributor g is used as a compressor for the Google semantics associated with the search terms and the associated NCD, called the *Normalized Google Distance (NGD)*, which is expressed as

$$NGD(s_1, s_2) = \frac{G(s_1 s_2) - min(G(s_1), G(s_2))}{max[C(s_1), C(s_2)]} = \frac{\max[\log f(s_1), \log f(s_2)] - \log f(s_1, s_2)}{\log N - \min[\log f(s_1), \log f(s_2)]}, \quad (22)$$

where $f(s_1)$ and $f(s_1, s_2)$ denotes the number of pages returned by Google search engine containing s_1 and both s_1,s_2.

6. *Hyperspace Analogue to Language (HAL)*: In this method, a set of words considered as *window* is analyzed, by passed over a corpus. A key assumption is made that co-occurring words of a window have a strength inversely proportional to the words which are separating them [31]. It produces a $n \times n$ matrix, where row and column contains the co-occurrence information of the words appearing before and following it and n is the size of the window. The resultant is a co-occurrence matrix and a vector can be formed of size $2n$ high dimensional space. It was found that during the experiment the co-occurrence value is 100 to 200 most variant vector elements, as reported in [31]. After the matrix construction, similarity can be measured between the word vectors. It was preferred that to measuring similarity two distance metrics: Euclidean if $(r = 2)$ and city-block if $(r = 1)$ metrics of Minkowski family of distance metrics is considered as follows:

$$distance = \sqrt[r]{\sum (|x_i - y_i|)^r}. \quad (23)$$

This is due to thefact that the vectors were normalized to a constant length by retaining small number of principal components and these metrics are sensitive to magnitude of the vector.

3.3 String-Based Similarity Measure

3.3.1 Character–Based Similarity Measure

1. *Longest Common Substring (LCS)*: This algorithm is used to find the longest substring of a string. It compares the two strings and find the similarity based on the longest common chain of characters. It can be measured as follows:

$$LCSubstr(S1, S2) = \max_{1 \le i \le m, 1 \le j \le n} LCSuff(S1_{1\ldots i}, S2_{1\ldots j}), \quad (24)$$

where m and n are the length of two string and $LCSuff$ is a function, which finds the of longest common suffixes of possible prefixes of $S1$ and $S2$.

2. *Damerau-Levenshtein*: It is a distance or string metric between two strings, which gives a number that required to transform one string into another. This transformation is done by insertion, deletion or substitution of a single character or a transposition of two adjacent characters [8].

3. *Jaro*: It is distance for similarity measure of two strings and the Jaro distance score is normalized to 0 means no similarity and 1 means an exact match. The *Jaro* distance d_j of two string is

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}, \quad (25)$$

where m is the number of matching characters and t is half the number of transpositions. The two characters of s_1 and s_2 are matched only when they are same and not far from $\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1$.

4. *Jaro–Winkler*: It is a semantic similarity measure of two strings. It is atype of edit distance and variant of *Jaro* distance metric and higher the Jaro–Winkler distance of two strings is more similar. This method is more useful for short strings such as person name.

The Jaro–Winkler distance d_w of two string s_1 and s_2 is:

$$d_w = d_j + (lp\,(1 - d_j))\,, \tag{26}$$

where p is the prefix scale gives ratings to the string that match from the beginning for a prefix length l and d_j is the *Jaro* distance [67].

5. *Needleman-Wunsch*: It is a type of dynamic algorithm used in Bioinformatics to align the protein sequences. It is also refereed as optimal matching algorithm and global alignment technique [38].
6. *Smith-Waterman*: It is a variation of Needleman-Wunsch algorithm and performs the local sequence alignment to measure the similarity between two strings or nucleotide or protein sequences. To measure the similarity it compares within the segments of the string and optimize the similarity. In general it is not used in large scale problem, because of its cubic computational complexity [59].
7. *n–gram*: It is a probabilistic language model used for predicting the next term in a sequence of $(n-1)$ terms or characters. *n–gram* model is used in various fields such as computational biology (for instance, biological sequence analysis), data compression, computational linguistic (for instance, statistical natural language processing) and computational theory. The main advantage of this model is *simplicity* and *scalability* [9].
8. *syntactic n–gram*: It is a modification of the n-gram language model, when the n-gram elements are taken not in the order as they appear in a text, but in the order that they appear in the corresponding syntactic tree [54, 53, 55] This approach allows introduciton of linguistic (syntactic) information into otherwise purely statisistical n-gram model. They can be applied in all tasks when traditional n-grams can be used [41, 58]). Obviously, in this case previous parsing is needed.

3.3.2 Term–Based Similarity Measure

1. *Block Distance*: It is also called city block distance or Snake distance or Manhattan distance or Manhattan length or L_1 distance or Taxicab distance [26]. It used to find the distance between two points. The taxicab distance, d_1, of two vectors p and q is

$$d_1\,(p,q) = \|p-q\|_1 = \sum_{i=1}^{n} |p_i - q_i|. \tag{27}$$

2. *Cosine Similarity*: It is a similarity measure of two non zero vectors of an inner product space, which finds cosine of the angle between them. Cosine of 0° is 1 and less than 1 for any other angle. The cosine similarity of two vectors have same orientation is 1 and vectors are in 90° have similarity 0. It is also used in data mining to finds the cohesion between them [63]. The cosine of two non zero vectors can be measured by Euclidean dot product.

$$a \cdot b = \|a\|\,\|b\| \cos\theta. \tag{28}$$

The cosine similarity $\cos(\theta)$ of two vectors A and B is

$$cos\,(\theta) = \frac{A \cdot B}{\|A\|\,\|B\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{29}$$

where A_i and B_i is the component of A and B.

3. *Soft Cosine Similarity*: It is a novel variant of the cosine similarity, when we take into account similarity of features in Vector Space Model [56].

$$soft\_cos\,(\theta) = \frac{\sum\limits_{i,j=1}^{n} s_{ij} A_i B_j}{\sqrt{\sum\limits_{i,j=1}^{n} s_{ij} A_i A_j}\sqrt{\sum\limits_{i,j=1}^{n} s_{ij} B_i B_j}}, \tag{30}$$

where s_{ij} is the value from the matrix of similarity between features i and j. Note that if this is diagonal matrix, i.e., the features are similar only to themselves, then soft cosine is equivalent to traditional cosine measure.

The similarity of features can be calculated for example, using WordNet similarity measures in case of words, or Levenshtein distance in case of strings or n-grams, or tree edit distance [57] in case of syntactic n-grams (n-grams constructed by following paths in syntactic trees [41, 58, 54]).

4. *Sorensen–Dice index*: It is also known as Sorensen index or Dice's coefficient and used to measure the similarity of two samples [14, 60]. It was adopted to find the presence or absence of data of two sets

$$QS = \frac{2\,|X \cap Y|}{|X| + |Y|}, \tag{31}$$

where $|X|$ and $|Y|$ is the number of elements in two sets and QS is the quotient of similarity and ranges between 0 and 1. When it is used to measure the similarity of strings $S1$ and $S2$ then coefficient can be calculated as bigrams as follows:

$$sim = \frac{2n_t}{n_{S1} + n_{S2}}, \tag{32}$$

where n_t is the bigram count of the strings and n_{S1}, n_{S2} is the number of bigrams in string $S1$ and $S2$.

5. *Euclidean Distance*: It is a distance between two points in Euclidean space. The Euclidean distance between two points s and t is

$$d\,(s,t) = d\,(t,s) = \sqrt{\sum_{i=1}^{n} (t_i - p_i)^2}. \tag{33}$$

6. *Jaccard Index*: It is also known as Jaccard similarity coefficient, is a statistics used to find the similarity and diversity between two finite sets [24]. It is defined as follows

$$J\,(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}. \tag{34}$$

The Jaccard distance is used to find the dissimilarity between two finite sets and is obtained by subtracting the Jaccard Index form 1 and refereed as d_J.

7. *Simple Matching Coefficient (SMC)*: It is a statistics used to find the similarity and diversity between two objects. It accepts the objects as a collection of n binary attributes and SMC of A and B is

$$SMC = \frac{Number\ of\ Matching\ Attributes}{Total\ no.\ of\ Attributes} = \frac{a_{00} + a_{11}}{a_{00} + a_{01} + a_{10} + a_{11}}, \tag{35}$$

where a_{00} represents total attributes as 0 in A and B; and a_{10} represents total attribute as 1's and 0's in A and B; and a_{01} represents total attributes as 0's and 1's in A and B; and a_{11} represents total attribute as 1's in A and B.

8. *Overlap Coefficient*: It is also called Szymkiewicz-Simpson coefficient, is also a similarity measure related to Jaccard index. The overlap between two sets is defined as

$$overlap(A,B) = \frac{|A \cap B|}{\min\left(|A|\,,|B|\right)}. \tag{36}$$

The overlap coefficient is equal to one when set A is a subset of set B.

4 Proposed Method

In this paper, a language model based semantic network has been proposed to find the semantic similarity between two English sentences. Among these two sentences one is considered as source, S, and other as target text, T. We assume that both, S and T, are syntactically and semantically correct. The proposed system can be brought down into the following stages:

1. In any language processing it is important to remove all the stop words before start any semantic similarity task. Initially all the stop words, have stored in a Java array and after that all the words of S and T is considered one after another for identification. Although stop words are most commonly used words but there is no universal list available for all language processing task[2]. These identified stop words are ignored during similarity stage.

[2] http://xpo6.com/list-of-english-stop-words/

2. In first step, Penn Treebank tag set [32] is used to label the words for part-of-speech (POS) information, which is most commonly used syntactic information. Further these identified tags and words are input to the system to generate the parse tree.
3. To generate the parse tree top down parsing is followed by considering its advantages over the bottom up parsing. For parsing all English grammatical role is considered. After that identified phrase structures is used to generate the top-down parse tree.
4. In this stage, a multi-stage (equal to level of the tree) undirected weighted graph is designed by considering the parse tree along with other statistical information found in the previous stages. Following characterises is considered for graph construction:
   (a) **Part-of-Speech:** All the stop words based on its POS information is not considered, when two words are found same in two parse tree at same level.
   (b) **Node Depth:** Starting from the root node S all possible paths are considered till the search ends with a word/concept at higher lever (i.e. leaf node) of the tree. The depth of any word is consider in the similarity measuring stage when a word is found in both the parse tree at same level and shares same POS tag.
   (c) **String Matching:** If any word is found in the parse tree of S and T, which possess *nnp* as POS tag then a weight value to the link is assigned if both the node are same.
5. After the completion of graph construction stage weight is measured between the nodes of two graphs. Assigning of weight is performed under the following condition:
   (a) if POS tag is found different of two nodes of same level then WordNet taxonomy relationship is considered. To calculate the information content i.e. weight w_i of the link the negative logarithm of the conditional probability (see Eq.(14)) as well as argument of information theory is considered.
   (b) if POS tag is different but strings are matched then two different weight values are calculated.

$$w_i^1 = sim\,(c_1, c_2) \tag{37}$$

and

$$w_i^2 = freq_{counts}\frac{c_i}{N}, \tag{38}$$

where c_1 and c_2 represents two concepts of two parse tree at same level. N represents as total number of words along with POS tag from a large text corpus and c_i represents total of class c. Finally, the maximum of w_i^1 and w_i^2 is considered for weight.

   (c) if no condition matched and phrase is identified as noun class and words are proper noun then no weight is measured for the link between the current node and proper noun node.
   (d) Finally, similarity is calculated as the minimum distance path while considering maximum weight of the link. After that, an average is calculated by summing of all weights of links starting form start node S till the leaf node.

5 Experimental Results for the Proposed Method

In order to evaluate the text similarity measure, pair of 50 sentences is taken from SemEval 2015 training dataset[3]. For this task, two different runs are conducted. For the first run, we consider WordNet taxonomy relationships and 0.46 similarity score is reported in this run. For this task WordNet version 2.0 is considered. In second run, we improved the similarity score using information content. For this task highest score is 0.78. In this method, we calculate the IC value by the combining of WordNet taxonomy and unigram language model, which outperforms the other methods reported in [22], [25] and [44].

[3] http://alt.qcri.org/semeval2015/task10/index.php?id=data-and-tools

6 Software for Semantic Similarity

1. *Semantic Measures Library (SML) and Toolkit*: It is a Java library and distributed under the open – source CeCILL license and designed for semantic measures. It can also used for computation and analysis purpose of semantic similarities between the term/ concepts defined in terminologies and ontologies [20]. It also supports comparison of entities (e.g. genes) annotated by concepts. Various ontological formats and specifications such as OBO, RDF and OWL also supported by this library. It also supports multi–threaded application for parallel computation. Based on SML an open–source toolkit also designed, which supports non–developers to use the functionalities through command prompt.

2. *WordNet–Similarity*: It is a WordNet based Perl module implements over the information found in the lexical database WordNet and able to find the semantic similarities and related measure. It supports list of measures like Rensik, Jiang-Conarth, Wu-Palmer, Banerjee-Pedersen, Patwardhan-Pedersen, Hirst-St.Onge, Leacock-Chodorow, and Lin. It has pre–computed pairwise similarity module for nouns and verbs [39].

3. *UMLS–Similarity*: Unified Medical Language System (UMLS), is a open source Perl module to find the semantic similarity and relatedness measure based on ontologies and terminologies found in medical domain. It gives a numerical value between a pair of medical concepts, which indicates the similarity between them [34].

4. *SEMILAR – A SEMantic SimiLARity Toolkit*: It is a single platform for user, researchers and developers to access the fully implemented Java based similarity methods. Its provides a GUI-based as well as a library to access the similarity methods. To access the similarity Methods offered in this tool starts from lexical to word–to–word similarity metrics to more sophisticated methods rely on unsupervised methods such as Latent Semantic Analysis and Latent Dirichlet Allocation to kernel-based methods. It also offers another tool called the SEMantic simILarity Annotation Tool (SEMILAT) for manual assessment and annotation by experts [49].

5. *DISCO Builder and API*: It is a Java based open source library to measure the similarity between words and phrases. It also allows to convert the Word2Vec or Glove vector files into DISCO word space index, which can be queried by DISCO API. DISCO[4] Builder is licensed under the Creative Commons Attribution-NonCommercial license and API is license under Apache.

6. *REST API*: It computes surface level semantic similarity between two texts using Cosine, Jaccard and Dice based similarity[5].

7. *TakeLab STS System*: It is a semantic text similarity system submitted as a evaluation exercise for task 6 in SemEval–2012[6].

7 Applications of Semantic Textual Similarity

1. *Biomedical Informatics*: To developed the biomedical ontologies namely the Gene Ontology we used the semantic similarity [40]. Similarity methods are mainly used to compare the genes and they can also used in other bio-entities [15].

2. *Geo–Informatics*: Similarity measure also used to find the similarities between geographical feature type ontologies. Several tools are available to do this task such as (i) The OSM Semantic Network used to compute the semantic similarity of tags in OpenStreetMap [7]. (ii) Similarity Calculator is used to find the similarity between two geographical concepts in the Geo–Net–PT ontology and (iii) SIM–DL similarity server computes the similarity between geographical feature type ontologies.

[4] http://www.linguatools.de/disco/disco-builder.html
[5] http://www.rxnlp.com/api-reference/text-similarity-api-reference/
[6] http://takelab.fer.hr/sts/

3. *Natural Language Processing*: It is field of Computer Science and linguistics. There are several fields where STS can play an important role directly or indirectly such as sentiment analysis, natural language understanding and machine translation.

8 Conclusion

In this survey, we explain four different measures for STS. We divide the String similarity measures into two categories as character based and term based. These two measures works on string sequences and character compositions, and measures the similarities and dissimilarities by calculating the distance between two strings or sets or vectors. We reported in total fourteen string similarity measures categorised into two groups. We also discussed three categories of topological methods such as node-based, edge-based and hybrid, the latter being a combination of node and edge-based. These topological studies are mainly used for finding the similarities and diversities between the terms and ontological concepts.

We also found that, node based approaches fully depends on the information content value between two nodes and distance based approaches depends on the depth of semantic network. On the other side, hybrid method works with weight value between child and parent nodes to find the similarity of two classes. In the statistical approach, we reported six different methods such as LSA, GLSA, ESA, PMI – IR, NGD, HAL. These statistical measures generates a vector space model from corpus and reduces the dimensionality of the vectors and finds the similarities and dissimilarities. Each cell of the matrix represents a frequency or weight value of a particular word in a paragraph or text passages.

Acknowledgements

This work presented here under the research project Grant No. YSS/2015/000988 and supported by the Department of Science & Technology (DST) and Science and Engi-neering Research Board (SERB), Govt. of India. Authors are also acknowledges the Department of Computer Science & Engineering of National Institute of Technology Mizoram, India for proving infrastructural facilities.

References

1. **Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., & Wiebe, J. (2014).** Semeval-2014 task 10: Multilingual semantic textual similarity. *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 81–91.
2. **Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., & Guo, W. (2013).** sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*, Citeseer.
3. **Agirre, E., Diab, M., Cer, D., & Gonzalez-Agirre, A. (2012).** Semeval-2012 task 6: A pilot on semantic textual similarity. *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pp. 385–393.
4. **Aliguliyev, R. M. (2009).** A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, Vol. 36, No. 4, pp. 7764–7772.
5. **Ando, R. K. (2000).** Latent semantic space: iterative scaling improves precision of interdocument similarity measurement. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 216–223.
6. **Bagrow, J. P. & Ben-Avraham, D. (2005).** On the Google-fame of scientists and other populations. *arXiv preprint physics/0504034*.
7. **Ballatore, A., Bertolotto, M., & Wilson, D. C. (2013).** Geographic knowledge extraction and semantic similarity in openstreetmap. *Knowledge and Information Systems*, Vol. 37, No. 1, pp. 61–81.
8. **Bard, G. V. (2007).** Spelling–error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string edit distance metric. *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*, Australian Computer Society, Inc., pp. 117–124.

9. **Barrón-Cedeno, A., Rosso, P., Agirre, E., & Labaka, G. (2010).** Plagiarism detection across distant language pairs. *Proceedings of the 23rd International Conference on Computational Linguistics*, Association for Computational Linguistics, pp. 37–45.
10. **Cilibrasi, R. L. & Vitanyi, P. M. (2007).** The google similarity distance. *IEEE Transactions on knowledge and data engineering*, Vol. 19, No. 3, pp. 370–383.
11. **Coelho, T. A., Calado, P. P., Souza, L. V., Ribeiro-Neto, B., & Muntz, R. (2004).** Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 4, pp. 408–417.
12. **Cohen, W. W. (2000).** Data integration using similarity joins and a word-based information representation language. *ACM Transactions on Information Systems (TOIS)*, Vol. 18, No. 3, pp. 288–321.
13. **Corley, C. & Mihalcea, R. (2005).** Measuring the semantic similarity of texts. *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, Association for Computational Linguistics, pp. 13–18.
14. **Dice, L. R. (1945).** Measures of the amount of ecologic association between species. *Ecology*, Vol. 26, No. 3, pp. 297–302.
15. **Ferreira, J. D. & Couto, F. M. (2010).** Semantic similarity for automatic classification of chemical compounds. *PLoS Comput Biol*, Vol. 6, No. 9, pp. e1000937.
16. **Fouss, F., Pirotte, A., Renders, J. M., & Saerens, M. (2007).** Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 3, pp. 355–369.
17. **Gabrilovich, E. & Markovitch, S. (2007).** Computing semantic relatedness using wikipedia-based explicit semantic analysis. *IJcAI*, volume 7, pp. 1606–1611.
18. **Ginsberg, A. (1993).** A unified approach to automatic indexing and information retrieval. *IEEE Expert: Intelligent Systems and Their Applications*, Vol. 8, No. 5, pp. 46–56.
19. **Halevy, A. Y., Ives, Z. G., Madhavan, J., Mork, P., Suciu, D., & Tatarinov, I. (2004).** The piazza peer data management system. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 7, pp. 787–798.
20. **Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2014).** The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. *Bioinformatics*, Vol. 30, No. 5, pp. 740–742.
21. **Ho Lee, J., Ho Kim, M., & Joon Lee, Y. (1993).** Information retrieval based on conceptual distance in is-a hierarchies. *Journal of documentation*, Vol. 49, No. 2, pp. 188–207.
22. **Hughes, T. & Ramage, D. (2007).** Lexical semantic relatedness with random graph walks. *EMNLP-CoNLL*, pp. 581–589.
23. **Islam, A. & Inkpen, D. (2008).** Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 2, No. 2, pp. 10.
24. **Jaccard, P. (1901).** *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz.
25. **Jiang, J. J. & Conrath, D. W. (1997).** Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.
26. **Krause, E. F. (1973).** Taxicab geometry. *The Mathematics Teacher*, Vol. 66, No. 8, pp. 695–706.
27. **Kucera, H. & Francis, W. N. (1982).** *Frequency analysis of English usage: Lexicon and grammar*. Boston: Houghton Mifflin.
28. **Landauer, T. K. & Dumais, S. T. (1997).** A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, Vol. 104, No. 2, pp. 211.
29. **Landauer, T. K., Foltz, P. W., & Laham, D. (1998).** An introduction to latent semantic analysis. *Discourse processes*, Vol. 25, pp. 259–284.
30. **Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006).** Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, Vol. 18, No. 8, pp. 1138–1150.
31. **Lund, K. & Burgess, C. (1996).** Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, Vol. 28, No. 2, pp. 203–208.
32. **Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993).** Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330.

33. **Matveeva, I., Farahat, A., & Royer, C. (2005).** Document representation with generalized latent semantic analysis. *Proceedings of the Conference 0n Research and Development in Information Retrieval (SIGIR 2005)*.

34. **McInnes, B. T., Pedersen, T., & Pakhomov, S. V. (2009).** Umls-interface and umls-similarity: open source software for measuring paths and semantic similarity. *AMIA Annual Symposium Proceedings*, volume 2009, American Medical Informatics Association, pp. 431.

35. **Mihalcea, R., Corley, C., & Strapparava, C. (2006).** Corpus-based and knowledge-based measures of text semantic similarity. *AAAI*, Vol. 6, pp. 775–780.

36. **Miller, G. A. (1995).** Wordnet: a lexical database for english. *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41.

37. **Mohler, M., Bunescu, R., & Mihalcea, R. (2011).** Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 752–762.

38. **Needleman, S. B. & Wunsch, C. D. (1970).** A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, Vol. 48, No. 3, pp. 443–453.

39. **Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004).** *WordNet:: Similarity: measuring the relatedness of concepts*. Association for Computational Linguistics.

40. **Pesquita, C., Faria, D., Falcao, A. O., Lord, P., & Couto, F. M. (2009).** Semantic similarity in biomedical ontologies. *PLoS comput biol*, Vol. 5, No. 7, pp. e1000443.

41. **Posadas-Durán, J., Markov, I., Gómez-Adorno, H., Sidorov, G., Batyrshin, I., Gelbukh, A., & Pichardo-Lagunas, O. (2015).** Syntactic n-grams as features for the author profiling task. *Working Notes Papers of the CLEF 2015 Evaluation Labs*, volume 1391 of *CEUR Workshop Proceedings*, CEUR.

42. **Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989).** Development and application of a metric on semantic nets. *IEEE transactions on systems, man, and cybernetics*, Vol. 19, No. 1, pp. 17–30.

43. **Ramage, D., Rafferty, A. N., & Manning, C. D. (2009).** Random walks for text semantic similarity. *Proceedings of the 2009 workshop on graph-based methods for natural language processing*, Association for Computational Linguistics, pp. 23–31.

44. **Resnik, P. (1992).** Wordnet and distributional analysis: A class-based approach to lexical discovery. *AAAI workshop on statistically-based natural language processing techniques*, pp. 56–64.

45. **Resnik, P. (1995).** Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

46. **Richardson, R. & Smeaton, A. (1995).** Using wordnet in a knowladge-based approach to information retrieval. Working Paper, CA-0395, School of Computer Applications, Dublin Sity University, Ireland.

47. **Rocchio, J. J. (1971).** *Relevance feedback in information retrieval*. Prentice-Hall, Englewood Cliffs NJ.

48. **Royer, C. (2007).** Term representation with generalized latent semantic analysis. *Recent Advances in Natural Language Processing IV: Selected papers from RANLP 2005*, Vol. 292, pp. 45.

49. **Rus, V., Lintean, M. C., Banjade, R., Niraula, N. B., & Stefanescu, D. (2013).** Semilar: The semantic similarity toolkit. *ACL (Conference System Demonstrations)*, pp. 163–168.

50. **Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997).** Automatic text structuring and summarization. *Information Processing & Management*, Vol. 33, No. 2, pp. 193–207.

51. **Schallehn, E., Sattler, K.-U., & Saake, G. (2004).** Efficient similarity-based operations for data integration. *Data & Knowledge Engineering, Elsevier*, Vol. 48, No. 3, pp. 361–387.

52. **Sheldon, R. (2002).** *A first course in probability*. Pearson Education India.

53. **Sidorov, G. (2013).** Non-continuous syntactic n-grams [in Spanish, abstract and examples in English]. *Polibits*, Vol. 48, pp. 67–75.

54. **Sidorov, G. (2013).** *Non-linear construction of n-grmas in computational lingusitics: Syntactic, filtered, and generalized n-grams*. SMIA, Mexico.

55. **Sidorov, G. (2014).** Should syntactic n-grams contain names of syntactic relations? *International Journal of Computational Linguistics and Applications*, Vol. 5, pp. 139–158.

56. **Sidorov, G., Gelbukh, A. F., Gómez-Adorno, H., & Pinto, D. (2014).** Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, Vol. 18, No. 3, pp. 491–504.
57. **Sidorov, G., Gómez-Adorno, H., Markov, I., Pinto, D., & Loya, N. (2015).** Computing text similarity using tree edit distance. *Proceedings of the Annual Conference of the North American Fuzzy Information processing Society and 5th World Conference on Soft Computing*, NAFIPS '15, pp. 1–4.
58. **Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Chanona-Hernández, L. (2013).** Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, Vol. 41, No. 3, pp. 853–860.
59. **Smith, T. F. & Waterman, M. S. (1981).** Identification of common molecular subsequences. *Journal of molecular biology*, Vol. 147, No. 1.
60. **Sørensen, T. (1948).** A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, Vol. 5, pp. 1–34.
61. **Steinberger, J. & Jezek, K. (2004).** Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM'04*, pp. 93–100.
62. **Sussna, M. (1993).** Word sense disambiguation for free-text indexing using a massive semantic network. *Proceedings of the second international conference on Information and knowledge management*, ACM, pp. 67–74.
63. **Tan, P.-N. et al. (2006).** *Introduction to data mining*. Pearson Education India.
64. **Tversky, A. (1977).** Features of similarity. *Psychological Review*, Vol. 84, No. 4, pp. 327.
65. **Šarić, F., Glavaš, G., Karan, M., Šnajder, J., & Bašić, B. D. (2012).** Takelab: Systems for measuring semantic text similarity. *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 441–448.
66. **Whan Kim, Y. & Kim, J. H. (1990).** A model of knowledge based information retrieval with hierarchical concept graph. *Journal of Documentation*, Vol. 46, No. 2, pp. 113–136.
67. **Winkler, W. E. (2006).** Overview of record linkage and current research directions. *Bureau of the Census*.
68. **Zobel, J. & Moffat, A. (1998).** Exploring the similarity space. *ACM SIGIR Forum*, Vol. 32, No. 1, pp. 18–34.

Goutam Majumder received his M.Tech degree in Computer Science & Engineering of Tripura University (A Central University), India as a first rank holder. He is currently Ph.D scholar and Assistant Professor at the Department of Computer Science & Engineering of the National Institute of Technology Mizoram. His working interest in image processing and natural language processing. He was also wroked as a reserach associate in Bio-Metrics Laboratory of Computer Science & Engineering Department of Tripura University (A Central University).

Partha Pakray received his Ph.D. degree in Computer Science and Engineering from the Jadavpur University, India. He is currently Head and Assistant Professor at the Department of Computer Science and Engineering of the National Institute of Technology Mizoram. He received fellowship from European Research Consortium for Informatics and Mathematics (ERCIM) for two times and worked at the Norwegian University of Science and Technology, Norway, and the Masaryk University, Czech Republic, as a postdoctoral fellow. He also worked at the Xerox Research Centre Europe (XRCE) as a research intern. He has published 50 research publications in various areas of NLP.

Alexander Gelbukh received his M.Sc. degree in Mathematics from the Lomonosov Moscow State University, Russia, and his Ph.D. in Computer Science from VINITI, Russia. He is currently a Research Professor and Head of the Natural Language Processing Laboratory of the Center for Computing Research (Centro de Investigación in Computación, CIC) of the Instituto Politécnico Nacional (IPN), Mexico. He is a former President of the Mexican Society of Artificial Intelligence

(SMIA), a Member of the Mexican Academy of Sciences, and a National Researcher of Mexico (SNI) at excellence level 2. He is author or coauthor of more than 500 research publications in natural language processing and artificial intelligence.

David Pinto received his PhD in computer science in the area of artificial intelligence and pattern recognition at the Polytechnic University of Valencia, Spain in 2008. At present he is a full time professor at the Faculty of Computer Science of the Benemérita Universidad Autónoma de Puebla (BUAP) where he leads the PhD program on Language & Knowledge Engineering. His areas of interest include clustering, information retrieval, crosslingual NLP tasks and computational linguistics in general. He has published more than 100 research publications in NLP and artificial intelligence.

Article received on 05/06/2016; accepted on 11/10/2016.
Corresponding author is Partha Pakray.