ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 2, Ver. 3 (April - June 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Graph Based Measure of Text Semantic Similarity Using WordNet as a Knowledge Base

[I]Majid Mohebbi, [II]Alireza Talebpour
[I,II]Dept. of Computer Engineering, Faculty of Electrical & Computer Engineering,
Shahid Beheshti University, Tehran, Iran
E-mail: ma.mohebbi@mail.sbu.ac.ir, talebpour@sbu.ac.ir

Abstract
A substantial part of the available information, is stored in text databases. Typically, only a minor fraction of available documents is appropriate for a user. Hence, to analyze and extract useful information from text documents, generation of the appropriate query document is difficult. This illustrates the importance of similarity of text documents. Several cases of lexical matching techniques offered to determine the similarity between documents that have been successful to a certain boundary and these methods are failing to find the semantic similarity between two texts. Therefore, the semantic similarity approaches were suggested, such as corpus-based methods and knowledge based methods e.g. WordNet based methods. This paper offers a new approach to the problem of text semantic similarity identification. In this work, we investigate a new approach of paraphrase identification in order to measuring the semantic similarity of texts. We present a Graph algorithm for Similarity identification that makes extensive use of word similarity information extracted from WordNet, also we implemented previously published method. Experiments performed on the Microsoft Research Paraphrase Corpus and we show our approach achieves appropriate performance.

Keywords
WordNet, Semantic Similarity, Similarity metric, Document Similarity, graph theory

I. Introduction

Natural Language Processing (NLP) or Natural Language Understanding (NLU) use of machinery approach for analyse, understand and generate human languages. Two main branches of NLP are Natural Language Analysis (NLA) and Natural Language Generation (NLG). Lexical, syntactic, semantic, pragmatic and morphological analysis of text are studied in NLA. Generation of eloquent multi-sentential or multi-paragraph response are studied in NLG [1]. Two approach in semantic similarity problem are paraphrase and bidirectional entailment. A paraphrase is a restatement of the meaning of a passage using other words. In NLG, paraphrases are an approach to increase variety of generated text [2]. Paraphrases take place at the word level, phrase level, sentence level or discourse level. Paraphrasing has at least three categories, Paraphrase Generation, Paraphrase Acquisition and Paraphrase Identification.

Paraphrase Generation (PG) is enumerated as a NLG problem is the task of generating alternative paraphrase text [3]. Paraphrase Acquisition or Paraphrase Extraction involves nominee paraphrases or extracting paraphrases from a large corpus [4]. Paraphrase Identification (PI) or Paraphrase Recognition (PR) or Paraphrase Detection (PD) is the task of recognizing the presence of paraphrase relationship at input texts. Textual entailment is the task of identifying, given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text [5]. A paraphrase can be considered as a bidirectional entailment relation namely text A is a paraphrase of text B if and only if A entails B and B entails A [2]. There are two main branches of PI, Unsupervised and Supervised learning. Unsupervised learning refers to the problem of trying to find hidden structure in unlabelled data. Supervised learning is the machine learning task of inferring a function from labelled training data [6]. The 'categories' in supervised learning are known but in unsupervised learning, system attempts to find appropriate 'categories'. For semantic similarity problem, in this article, we focus on sentential paraphrases by unsupervised approach. The following is an introduction on similarity of texts problem. Similarity between two candidate texts typically was to use a simple lexical matching approach, and produce a similarity score based on the number of lexical units that take place in both input segments. Stemming, stop-word removal, part-of-speech tagging, longest subsequence matching, as well as various weighting and normalization factors have considered to improvement to this simple method [7]. These methods, while successful to a particular degree, will fail to recognize the similarity between sentences which use different, but synonymous, words to carry the same meaning. For determining the similarity of a pair of words, several methods are available, including several techniques based on WordNet, e.g. Leacock and Chodorow, 1998; Wu and Palmer, 1994; Resnik, 1995; Lesk 1986; Lin 1998; Jiang & Conrath 1997. For text semantic similarity, perhaps the most widely used approaches is the latent semantic analysis method [8]. However, due to the complexity and computational cost, LSA has not been used on a large scale, also the algorithm does not allow for any deep insights into why some terms are selected as similar during the singular value decomposition process. A related work consists of unsupervised methods for paraphrase identification, such as methods that Mihalcea et al [9] described for paraphrase recognition and Semantic similarity matrix is described by Fernando and Stevenson [10] which made use of WordNet based methods. While these approaches have the potential for high precision on many examples, improper selection of specific similarity weight are often insurmountable. Ramage, et al [11] present an algorithm for Text Semantic Similarity, coining the name "RandomWalks for Text Semantic Similarity" for his work. This paper presents a new method, the Graph based approach. This approach uses Graph algorithm to find the similarity of two text segments, but a key difference is that Special word to word similarities are taken into account, not just the maximal similarities or not all similarities between the sentences as in the methods proposed in Mihalcea et al [9] and Fernando and Stevenson [10]. We show performance of our approach via evaluating on a paraphrase recognition task. The rest of this paper is organized as follows: Section 2 reviews existing similarity measures. In Section 3 we offer the new similarity measure based on the graph-based

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

measure. Section 4 introduces the DataSet. Experiments and results are described in Section 5. Section 6 gives our conclusions.

II. Previous Approaches

Madnani et al [12] re-examined the idea that automatic metrics used for evaluating translation quality for the task of paraphrase recognition. They employed 8 different Machine Translation metrics for identifying Paraphrases. They found that a meta-classifier trained using only MT metrics obtained proper results.
Zia and Wasif [13] offered approach of paraphrase identification using semantic heuristic features. In this approach the POS tagger is performed and closed-class words is removed, after pre-processing step, the feature set is defined. Features was extracted for each sentence pair, afterwards Machine Learning phase is done. Moreover, a detailed misclassification analysis has been carried out to provide an insight into the syntactic structure of corpus causing misclassifications.
Rajkumar and Chitra [14] offered a neural network classifier for recognizing paraphrases. A combination of lexical, syntactic and semantic features has been used to construct feature vector to train a Back Propagation network. For Feature Extraction, approaches such as modified string edit distance, the Jiang and Conrath measure, skip-grams with skip distance k as 4 and adapted BLEU metric were used. Moreover dependency tree edit distance, Parts of Speech enhanced Position Error Rate and negations handling were used to construct feature vector. Once the network is trained its performance can be evaluated on test data.
Rus et al [2] offered a graph subsumption approach for Paraphrase Recognition. The input sentences are mapped to graph structures and subsumption is detected by evaluating graph isomorphism. The entailment score for Text A with respect to Text B and B with respect to A have been averaged to determine whether A and B are paraphrases.
The approach developed by Mihalcea et al [9], surpasses simple lexical matching. To estimate the semantic similarity of the sentence pairs, Word-to-word similarity measures and a word specificity measure are used. The approach offers following scoring function:

$$Sim(T_1,T_2)=\frac{1}{2}\left(\frac{\sum_{w\in\{T_1\}}(\max Sim(w,T_2)*idf(w))}{\sum_{w\in\{T_1\}}idf(w)}+\frac{\sum_{w\in\{T_2\}}(\max Sim(w,T_1)*idf(w))}{\sum_{w\in\{T_2\}}idf(w)}\right) \quad (1)$$

Where $\max Sim(w,T)$ is the maximum similarity score between word $\underline{w}$ and words in T according to one of the word to word similarity measures, and $idf(w)$ is the inverse document frequency of the word. A threshold of 0.5 was used for classification. A score above the threshold was labelled as a paraphrase otherwise as not paraphrase.
The main idea in the approach proposed by Fernando and Stevenson [10] uses the matrix similarity approach to find the similarity of two text segments, but a key difference is that all word to word similarities are taken into account, not just the maximal similarities between the sentences as in the method proposed in Mihalcea et al [9]. The following scoring function was used for computing similarity between sentences:

$$Sim(\vec{a},\vec{b})=\frac{\vec{a}W\vec{b}}{|\vec{a}||\vec{b}|} \quad (2)$$

Where W is a semantic similarity matrix containing information about the similarity of word pairs.
The approach developed by Ramage et al [11] compare the distribution each text induces when used as the seed of a random walk over a graph constructed from WordNet and corpus statistics. Their algorithm aggregates local relatedness information via a random walk over a graph constructed from an underlying lexical resource. The stationary distribution of the graph walk forms a "semantic signature" that can be compared to another such distribution to get a relatedness score for texts [11].

III. Graph Based Approach

Number of previous unsupervised works have shown that similarity measures is still limited by the fact that only the most similar or All similar word for the other sentence is taken into account.
In this paper, we explore an unsupervised knowledge-based method for measuring the semantic similarity of texts that specific word to word similarities are taken into account, not just the maximal similarities or all similarities between the sentences. In the following, we present our algorithm with an example.
In the MSR Paraphrase Corpus [15], the paraphrase pair "191346-191536", is assessed at dissimilar.
First sentence is: "Worldwide, 7,183 SARS cases and 514 deaths have been reported in 30 countries".
Second sentence is: "Taiwan reported 22 new cases, for a total of 360 with 13 deaths".
For a given pair of text segments, we begin by producing sets of open-class words, with a distinct set created for nouns, verbs, adjectives - adverbs - cardinals. Next, we try to determine similarity of pairs of words across the sets corresponding to the same open-class in the two text segments. We enforce the "same word-class" restriction to all the word-to-word similarity measures. For nouns and verbs, we use a measure of semantic similarity based on WordNet, while for the other word classes we use lexical matching.
We execute a part-of-speech tagging on a sentence using Stanford tagger [16].
We construct a bipartite graph $G=\{X,Y,E\}$ with vertices $X\cup Y$ that X shows words associated with a one class of first sentence and Y shows words associated with a same class of second sentence and edges E extracted from WordNet 3.0 and an edge is placed between every two congener classes. No edge is placed between two incongruous classes.
Fig. 1 shows constructed graph for two candidate sentences by wup measure values of WordNet::Similarity package [17] to determine the similarity of pairs of words across the same segment in the two text. Zero weight Edges are not drawn and there is no edge between incongruous classes of two sentences.
The approach proposed by Mihalcea et al [9] selects the maximum similarity for each word, for example at Fig. 1, for the noun 'total' in second sentence, this approach finds the most similar word that is related to noun 'cases'. E.g. weighting by this approach for selected pair of sentences, gives that these two sentences are as paraphrase, that is incorrect detect.
At semantic similarity matrix [10] was considered all similarity values to complete the similarity matrix, compared to Mihalcea et

al [9] approach. These approach by selecting additional weights that be affecting the accuracy of system, increases computing time.

We intend to apply a graph approach to select only the specific weights (edges) similarity values of pairs of words.

For provision of our approach, first we introduce OM algorithm in graph theory.

Optimal Matching (OM) [18] is classical problem in graph theory. Let $G = \{X, Y, E\}$ be a bipartite graph, where $X = \{x_1, x_2, ..., x_m\}$ and $Y = \{y_1, y_2, ..., y_n\}$ are the partitions, $V = X \bigcup Y$ is the vertex set, and $E = \{e_{ij}\}$ is the edge set.

OM is to find the matching M of G that has largest total weight and a subset of the edges with the property that no two edges of M share the same node.

Fig. 2 shows the result of OM algorithm for given the weighted bipartite graph G. The edges that are selected by OM have no two edges that is shared with the same nodes and the edges have largest total weight.

We propose a new similarity measure based on optimal matching concept. We do not intend to find the matching of that has largest total weight, we change this feature. We select the edge with the largest weight respectively and each node in V appears in at most one edge in M. We are coining the name "Extended OM" or "EOM" for this algorithm.

It should be noted that EOM will apply separately to each pair of class of nouns, verbs, adjectives - adverbs - cardinals. In other words, there would be no edge between incongruous classes, even with zero weight.

To applying the EOM algorithm to calculate the similarity between two sentences, in order to select values of similarity, we also consider the edges with zero weight across the same class that they will be used to choose by EOM. The cause is an impact of the words that have no resemblance to the corresponding class of another sentence on the amount of similarity value. These words have increased length sentence, In other words, in general similarity have been reduced.

Fig. 3 shows the implied edges with a dash grey line. The grey edges have zero weight. There will be a chance to choose the implied edges By EOM.

Now, Edges are selected by EOM. Fig. 4 shows the selected edges.

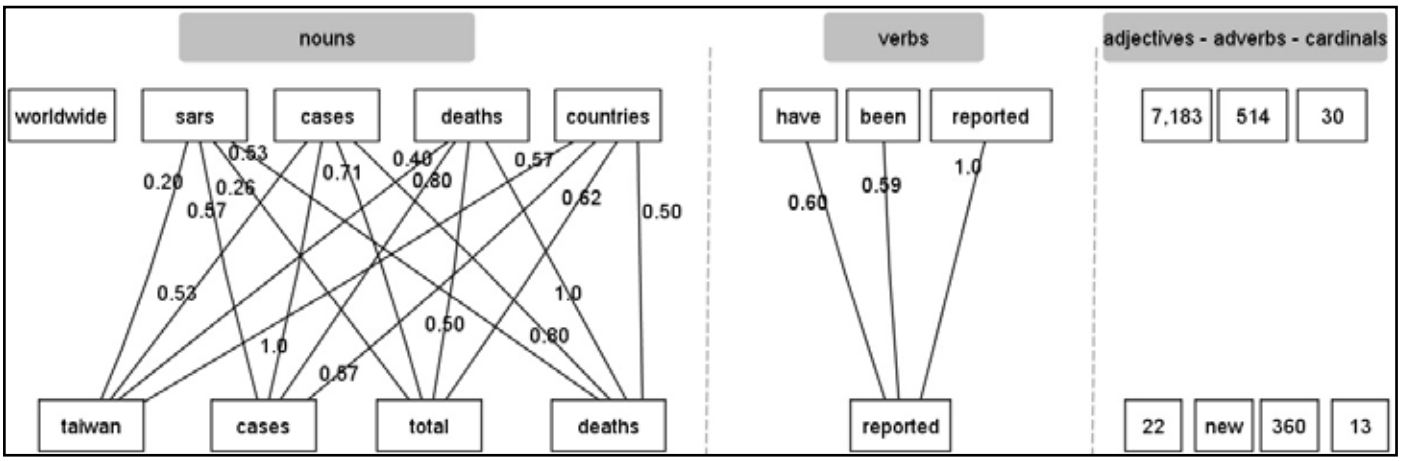

Fig. 1: Constructed Graph by wup measure Values for pairs of words. First row shows first sentence elements and second row shows second sentence elements. Zero weight Edges are not drawn and there is no edge between incongruous classes of two sentences.

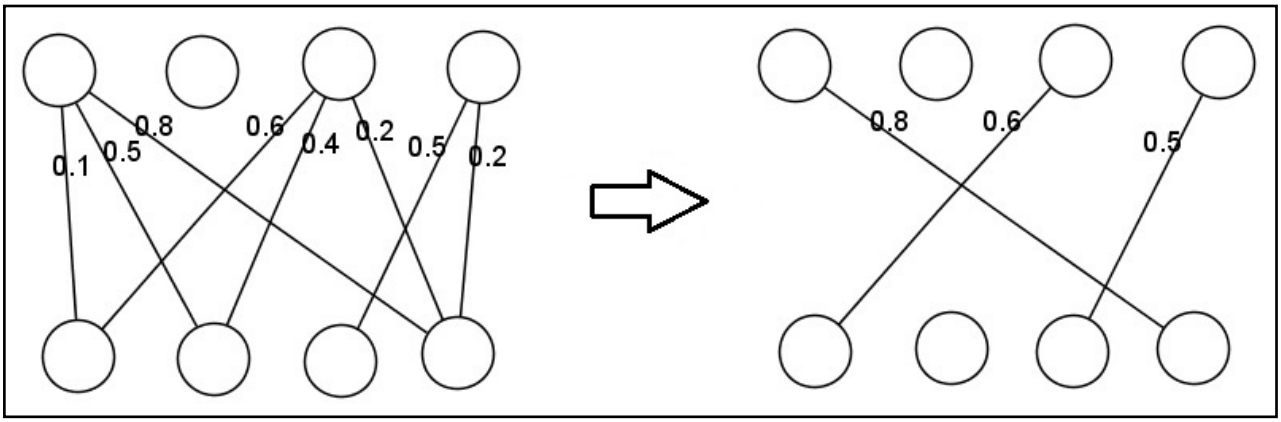

Fig. 2: Optimal matching.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

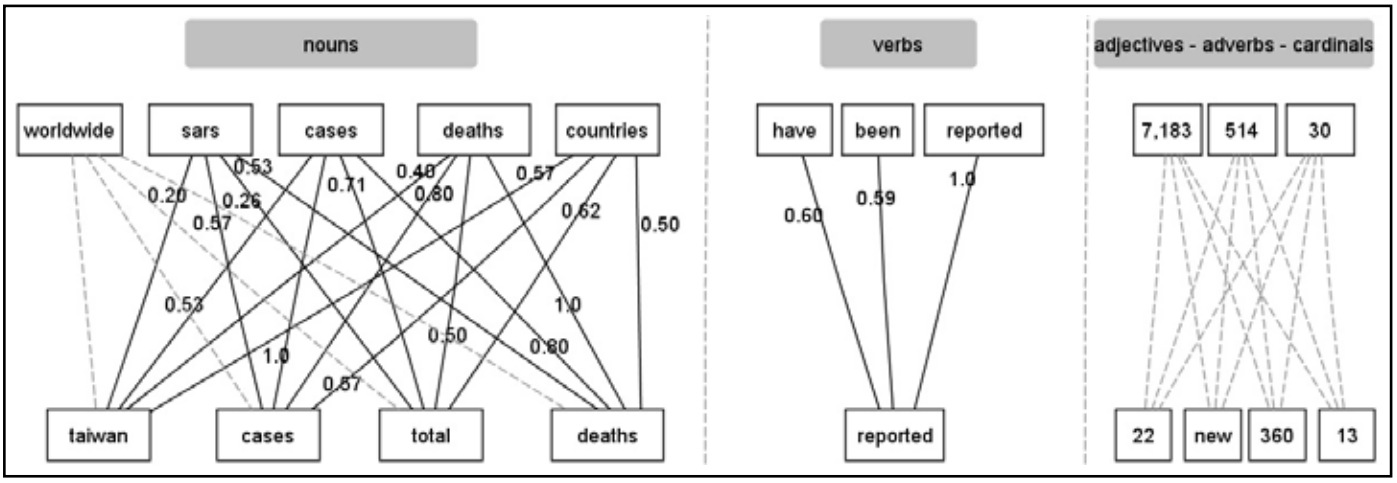

Fig. 3: wup measure value for pairs of words along with dash grey lines (the implied edges have zero weight).

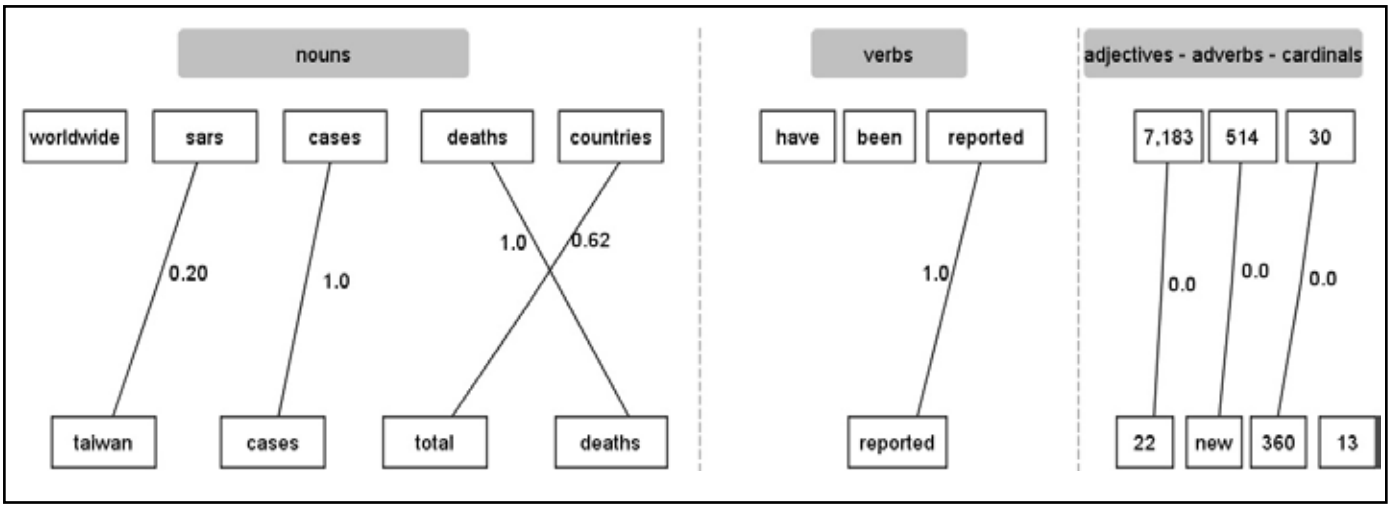

Fig. 4: Result of our approach - selected edges by EOM algorithm.

The difference between our approach and Mihalcea et al [9] approach is that Mihalcea et al [9] approach selects maximum value of similarity for word "sars", Namely 0.57, but in our approach 0.2 is seleted, that is unaffected by the choice of selecting the previous edges.

Using the weights of selected edges and number of nodes, the similarity between the two texts is determined using the following scoring function:

$$Sim(T_1,T_2) = \frac{\sum weight\ of\ Selection\ Edges}{\frac{1}{2}(Number\ of\ nods(T_1) + Number\ of\ nods(T_2))} \quad (3)$$

For example, for two candidate sentence form the dataset that Fig. 4 has shown selected edges, by using the metric shown in formula 3, the similarity between sentences is:

$$Sim(T_1,T_2) = \frac{0.20+1.0+1.0+0.62+1.0+0+0+0}{\frac{1}{2}(11+9)} = 0.382$$

We use a threshold of 0.56 for classification; a score below the threshold was classified as non-similar sentence otherwise as similar (paraphrase).

In the following, we present other two kinds of our algorithm, Second type and third type. We take into account the specificity of words, in order that we give a higher weight to the similarity measured between specific two words, and give less importance to the similarity calculated between generic concepts.

For determining the specificity of a word, we use the inverse document frequency (IDF) [19], defined as the total number of documents in the corpus divided by the total number of documents including that word. We use "BNC database and word frequency lists" by Adam Kilgarriff for document frequency counts for experiments reported here.

In the second type algorithm, for each edge, we multiply the edge weights by average IDF of two nodes of an edge, afterwards we run our first algorithm (EOM). We are coining the name "EOM before" for this algorithm. Feature of this algorithm is combining the word similarities and their specificity.

The similarity for EOM before is determined using the following scoring function:

$$Sim_{before}(T_1,T_2) = \frac{\sum idfweighted\ of\ Selection\ Edges}{\frac{1}{2}(\sum idf\ of\ nods(T_1) + \sum idf\ of\ nods(T_2))} \quad (4)$$

Using equation 4, we get the semantic similarity of the two candidate sentence as 0.484, i.e. correct diagnosis (not paraphrase).

In the third type algorithm, first we run EOM, afterwards for selected edges, we multiply the selected edge weights by average IDF of two nodes of an edge. We are coining the name "EOM after" for this algorithm. The similarity for EOM after is determined using equation 4.

Using Equation 4, we get the semantic similarity of the two texts

for EOM after as 0.386. For this example, approach proposed by Mihalcea et al [9] get score as paraphrase, i.e. wrong detection.

1. Computing Lexical Similarity

To quantify the degree of semantically relation of two words, we use six measures. Leacock & Chodorow [20], Wu & Palmer [21], Resnik [22], Lin [23], and Jiang & Conrath [24] which use only information about the is-a hierarchy to determine the similarity of the concepts. Metric Lesk [25] uses additional information apart from hypernymy to measure the similarity of the two concepts. We use the WordNet-based implementation of these metrics available in the WordNet::Similarity package [17].

The lesk metric [25] measures the overlap between the corresponding definitions, as provided by a dictionary and also concepts directly related via relations such as hypernyms and meronyms.

The lch metric [20] finding the path length between two nodes in the is-a hierarchy. The similarity is computed as:

$$Sim_{lch} = -\log \frac{length}{2 * D} \quad (5)$$

Where $length$ is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the taxonomy.

The wup metric [21] computes the depth of two concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), LCS is defined for $concept1$ and $concept2$ in a is-a hierarchy as the most specific node which both nodes share as an ancestor. The similarity between nodes $concept1$ and $concept2$ is:

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \quad (6)$$

The res metric [22] calculate the information content (IC) of the LCS of two concepts:

$$Sim_{res} = IC(LCS) \quad (7)$$

Where IC is defined as:

$$IC(c) = -\log P(c) \quad (8)$$

Where $P(c)$ is the probability of finding c in a large corpus.

The lin metric [23] normalise resnik measure by using the information content of the two nodes themselves:

$$Sim_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)} \quad (9)$$

The jcn metric [24]:

$$Sim_{jcn} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(LCS)} \quad (10)$$

Only the score of lin and wup measures is between 0 and 1. The remaining measures are normalized range of 0–1 by dividing the similarity score provided by a given measure with the possible maximum score for that measure.

IV. The Dataset

The Microsoft Research Paraphrase Corpus has been used throughout our experiments. It is the result of an effort to construct a large scale paraphrase corpus for generic purposes [15]. It consists of 5,801 sentence pairs extracted from online newswire text, in which 3,900 are tagged as true paraphrases by human judges. The data have been arbitrarily split into a training set containing 4076 examples and a test set containing 1725 examples. The human judges agreement was measured at approximately 83%, which can be considered as upper accuracy of automatic methods. Our algorithm can be used as unsupervised or supervised. At unsupervised experimental setting, we only use the test data in the experiments and for each pair in the test set, we evaluate our algorithm, and we use threshold of 0.56.

V. Evaluation and Results

The goal of our evaluation is to show accuracy, precision, recall and F_measure of our system, calculated concerning the true and false marked values in the test data. Accuracy is how close a generated value is to the actual value. Precision is how close the generated values are to each other and is the fraction of retrieved occurrences that are relevant. Recall is the fraction of relevant occurrences that are retrieved. High recall means that an algorithm generated most of the relevant results, while high precision means that an algorithm generated more relevant results than irrelevant. F_measure is a measure that combines precision and recall. We compare results of our system as unsupervised algorithms with other unsupervised approaches. Table. 1 shows the results obtained of our algorithms in the unsupervised setting using threshold of 0.56.

Table. 1: Experiment results of our algorithms on MSR Paraphrase Corpus by using threshold of 0.56.

	Metric	Acc.	Prec.	Rec.	F
	Semantic similarity (knowledge-based)				
EOM	J & C	72.52	74.91	88.23	81.02
	L & C	71.83	71.04	97.30	82.12
	Lesk	71.88	75.66	85.09	80.10
	Lin	72.81	72.97	93.90	82.12
	W & P	71.13	70.45	97.47	81.78
	Resnik	73.16	73.20	94.07	82.33
E O M after	J & C	70.90	75.29	83.70	79.27
	L & C	71.71	72.52	92.50	81.30
	Lesk	69.74	75.51	80.65	77.99
	Lin	71.19	73.93	87.53	80.16
	W & P	70.90	71.69	92.94	80.94
	Resnik	70.96	73.75	87.45	80.02
E O M before	J & C	70.96	75.04	84.39	79.44
	L & C	70.96	71.39	93.98	81.14
	Lesk	69.68	75.24	81.08	78.05
	Lin	71.65	73.47	89.80	80.82
	W & P	70.49	70.77	94.77	81.03
	Resnik	65.28	75.51	70.71	73.03

As we showed the experiment results of our three approach in Table 1, this result indicates that "EOM" offers better results than "EOM after" and "EOM before" approach. The reason is that only open-class words have evaluated by our algorithm and closed-class words were removed. Because the use of valence of the words has not the desired effect. Hence, we compared the

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

results of EOM approach to other approaches.

For having fair judgment result, we generate result of Mihalcea et al's measure [9] by using WordNet3.0. Hence we implement Mihalcea et al's measure then evaluate it. For obtained results, we use threshold of 0.50 as mentioned by Mihalcea et al [9].

The comparison between results obtained in table 2 and the results reported in Mihalcea et al [9], we observed an increase in accuracy by applying WordNet3.0.

Table 2 shows comparison between the results of our system and Mihalcea et al's measure and a representative corpus-based measure and Baselines as reported in Mihalcea et al [9]. Also in table 2, the results reported in Mihalcea et al [9] associated with six metric are shown. Comparison shows our approach to outperform Mihalcea et al [9]. Peer to peer comparing results of six metrics at accuracy, show superiority of our measure.

Table 3 shows the results of our system (EOM algorithm) and a representative subset of those reported in Ramage et al [11] that be used version 3.0 of WordNet. We observed our approach outperform Random GraphWalk approach.

Table. 2: Experiment results of our method and Mihlcea et al's approach.

	Metric	Acc.	Prec.	Rec.	F
	Semantic similarity (knowledge-based)				
EOM	J & C	72.52	74.91	88.23	81.02
	L & C	71.83	71.04	97.30	82.12
	Lesk	71.88	75.66	85.09	80.10
	Lin	72.81	72.97	93.90	82.12
	W & P	71.13	70.45	97.47	81.78
	Resnik	73.16	73.20	94.07	82.33
Mihlcea et al's approach by using WordNet 3.0	J & C	70.38	71.46	92.33	80.56
	L & C	69.10	68.81	97.91	80.82
	Lesk	69.91	71.72	90.41	79.98
	Lin	70.20	70.17	95.99	81.08
	W & P	69.28	68.80	98.43	80.99
	Resnik	69.80	69.81	96.16	80.89
Mihlcea et al's approach [9]	J & C[9]	69.3	72.2	87.1	79.0
	L & C[9]	69.5	72.4	87.0	79.0
	Lesk[9]	69.3	72.4	86.6	78.9
	Lin[9]	69.3	71.6	88.7	79.2
	W & P[9]	69.0	70.2	92.1	80.0
	Resnik[9]	69.0	69.0	96.4	80.4
	Combined [9]	70.3	69.6	97.7	81.3
Mihlcea et al's measure [9]	Semantic similarity (corpus-based)				
	PMI-IR[9]	69.9	70.2	95.2	81.0
	LSA[9]	68.4	69.7	95.2	80.5
	Baselines				
	Vector-based[9]	65.4	71.6	79.5	75.3
	Random[9]	51.3	68.3	50.0	57.8

Table. 3: Experiment results of our method and Random GraphWalk approach.

	Metric	Acc.	F
EOM	J & C	72.52	81.02
	L & C	71.83	82.12
	Lesk	71.88	80.10
	Lin	72.81	82.12
	W & P	71.13	81.78
	Resnik	73.16	82.33
Random GraphWalk[11]	Walk (Cosine) [11]	68.7	78.7
	Walk (Dice)[11]	70.8	80.1
	Walk (JS)[11]	68.8	80.5

Semantic matrix [10] approach has presented unsupervised algorithms. According to author claim, "For each similarity metric the training part of the MSRPC was used to find the classification threshold for the similarity score which maximized accuracy.", this approach use a supervised setting, where the optimal threshold of similarity metrics are concluded through learning on training data. Nonetheless for the proposed algorithm achieved maximum accuracy is 73.16%, whereas semantic matrix [10] approach achieved maximum accuracy is 74.1%, but more time is needed by semantic matrix [10] approach because more word pair are needed to calculate the similarity matrix, and consequently more time for calculate the similarity values is needed.

VI. Conclusion

In this paper, we presented a new approach for usage of graph theory concepts for computing text semantic similarity. For computing semantic relatedness, we offer WordNet-based semantic similarity measures.

Using Extended Optimal Matching algorithm for selecting specific edges, we obtained appropriate results. By selecting specific edges, only specific weight of similarity are selected for each pair of words. Our proposed algorithm do not attempt to find the max similarity for each words and do not used all similarity values, rather it selects the certain weights (edges), according to previous selections.

By using specificity of words, we present other two kinds of first proposed algorithm. Results indicated that first algorithm to outperform the other two algorithms.

We evaluated our system on the Microsoft Research Paraphrase Corpus and achieve appropriate performance.

References

[1] *N. Indurkhya and F. J. Damerau, Handbook of Natural Language Processing, 2nd ed., CRC Press, Boca Raton, 2010.*

[2] *V. Rus, P. M. McCarthy, M. C. Lintean, D. S. McNamara, and A. C. Graesser, "Paraphrase identification with lexico-syntactic graph subsumption," Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference, pp. 201-206, 2008, Coconut Grove, Florida.*

[3] *S. Wubben, A. Van den Bosch, and E. Krahmer, "Paraphrase*

generation as monolingual translation: data and evaluation," Proceedings of the 6th International Natural Language Generation Conference (INLG 2010), pp. 203-207, 2010, Stroudsburg, PA, USA.

[4] R. Bhagat, E. Hovy, and S. Patwardhan, "Acquiring paraphrases from text corpora," Proceedings of the 5th international Conference on Knowledge Capture, pp. 161-168, 2009, New York, USA.

[5] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment, Vol. 3944, pp. 177-190, 2006.

[6] Unsupervised learning, from Wikipedia, the free encyclopedia,[Online], http://en.wikipedia.org/wiki/Unsupervised_learning

[7] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," Information Processing & Management, Vol. 24, No. 5, pp. 513-523, 1988.

[8] T. K. Landauer, P. W. Foltz, and D. Laham, "An Introduction to latent semantic analysis," Discourse Processes, Vol. 25, No. 2-3, pp. 259-284, 1998.

[9] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity," Proceedings of the American Association for Artificial Intelligence (AAAI 2006), 2006, Boston.

[10] S. Fernando and M. Stevenson, "A semantic similarity approach to paraphrase detection," Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics, 2008.

[11] D. Ramage, A. N. Rafferty, and C. D. Manning, "Random walks for text semantic similarity," Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4), pp. 23-31, 2009, Stroudsburg.

[12] N. Madnani, J. Tetreault, and M. Chodorow, "Re-examining machine translation metrics for paraphrase identification," Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 182–190, 2012, Montr´eal, Canada.

[13] U. Zia and A. Wasif, "Paraphrase Identification using Semantic Heuristic Features," Research Journal of Applied Sciences, Engineering and Technology, Vol. 4, No. 22, pp. 4894-4904, 2012.

[14] A. Rajkumar and A. Chitra, "Paraphrase recognition using neural network classification," International Journal of Computer Application, Vol. 1, No. 29, 2010.

[15] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources," Proceedings of the 20th international conference on Computational Linguistics (COLING '04), pp. 350, 2004, Morristown, NJ, USA.

[16] K. Toutanova, et al., "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," Proceedings of Human Language Technology (NAACL '03), pp. 252-259, 2003, Edmonton.

[17] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity: Measuring the Relatedness of Concepts," Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), pp. 1024–1025, 2004, San Jose, CA.

[18] X. Wan, "A novel document similarity measure based on earth mover's distance," Information Sciences, Vol. 177, No. 18, pp. 3718–3730, 2007.

[19] K. Sparck-Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, Vol. 28, No. 1, pp. 11–21, 1972.

[20] C. Leacock and M. Chodorow, "Combining local context and WordNet sense similarity for word sense identification," In WordNet, An Electronic Lexical Database. The MIT Press. 1998.

[21] Z. Wu and M. Palmer, "Verb semantics and lexical selection," Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 133-138, 1994.

[22] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), 1995.

[23] D. Lin, "An information-theoretic definition of similarity," Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98), pp. 296–304, 1998, San Francisco, CA, USA.

[24] J. J. Jiang, and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," Proceedings of the International Conference on Research in Computational Linguistics. (ROCLING X), 1997, Taiwan.

[25] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," Proceedings of the 5th annual international conference on Systems documentation (SIGDOC '86), pp. 24-26, 1986, New York.

Author's Profile and Image

Majid Mohebbi received the MSc degree in software engineering from Shahid Beheshti University in 2013, Tehran, Iran. His research interests include Semantic Similarity and NLP.

Alireza Talebpour received his MSc degree in Artificial Intelligence and PhD degrees in Image Processing from University of Surrey, Guildford, United Kingdom. His research interests include image processing and pattern recognition, intelligent methods for classification of Massive Data.