4-2011

Semantic Relevance Analysis of Subject-Predicate-Object (SPO) Triples

Ranjana Kumar
University of Nebraska at Omaha

Semantic Relevance Analysis of Subject-Predicate-Object (SPO) Triples

A Thesis Presented to the

Department of Computer Science and the Faculty of the Graduate College

University of Nebraska

In Partial Fulfillment of the Requirements for the Degree

Masters of Science

University of Nebraska at Omaha

By

Ranjana Kumar

April, 2011

Supervisory Committee

Qiuming Zhu, Chair

Robin Gandhi

William Mahoney

Parvathi Chundi

Advisor: Qiuming Zhu

Semantic Relevance Analysis of Subject-Predicate-Object (SPO) Triples

Ranjana Kumar, MS

University of Nebraska at Omaha, 2011

Advisor: Qiuming Zhu

Abstract

The goal of this thesis is to explore and integrate several existing measurements for ranking the relevance of a set of subject-predicate-object (SPO) triples to a given concept. As we are inundated with information from multiple sources on the World-Wide-Web, SPO similarity measures play a progressively important role in information extraction, information retrieval, document clustering and ontology learning. This thesis is applied in the Cyber Security Domain for identifying and understanding the factors and elements of sociopolitical events relevant to cyberattacks. Our efforts are towards developing an algorithm that begins with an analysis of news articles by taking into account the semantic information and word order information in the SPOs extracted from the articles. The semantic cohesiveness of a user provided concept and the extracted SPOs will then be calculated using semantic similarity measures derived from 1) structured lexical databases; and 2) our own corpus statistics. The use of a lexical database will enable our method to model human common sense knowledge, while the incorporation of our own corpus statistics allows our method to be adaptable to the Cyber Security domain. The model can be extended to other domains by simply changing the local corpus. The integration of different measures will help us triangulate the ranking of SPOs from multiple dimensions of semantic cohesiveness. Our results are compared to rankings gathered from surveys of human users, where each respondent ranks a list of SPO based on their common knowledge and understanding of the relevance evaluations to a given concept. The comparison demonstrates that our integrated SPO similarity ranking scheme closely reflects the human common sense knowledge in a specific domain it addresses.

***Keywords**: Cyber attack, Semantic Similarity, Subject Predicate Object, Knowledge based Measure, Corpus Based Measure, Statistical Analysis, Local Corpus*

Acknowledgements

In the first place I would like to thank my adviser Dr. Quiming Zhu, who has supported me throughout my thesis work with his patient and valuable guidance. Above all and the most needed, he provided me steadfast encouragement and support in various ways from the very early stage of in master program, and this research.

I gratefully acknowledge and would like to thank Dr. Robin Gandhi for his crucial contribution of advice and supervision, which made him the backbone of this thesis. His involvement and his originality have triggered and nourished my intellectual maturity that I will benefit from for a long time to come.

I would like to thank Dr. Bill Mahoney for all his support, constructive comments and valuable suggestions for my thesis work.

I would also like to thank my committee member Dr. Parvathi Chundi. The valuable knowledge I have gained through her classes has proved to be very beneficial for the work of this thesis and have made my graduate studies very rewarding.

My deepest gratitude goes to my family especially to my husband Anjani Kumar for his constant love and support throughout my study and my life; this thesis and study would have been simply impossible without his support. I really want to thank him for all his motivation and encouragement in all the areas of my life.

Finally, I wish to thank to my Kewi Group and all others who directly or indirectly helped me in completing this thesis. This work would not have been possible without all of your support and love. Thank you.

Table of Contents

List of Figures

List of Equations

List of Tables

1 Introduction

An ontology of a certain domain explicitly identifies its terminology (domain vocabulary), the essential concepts in that domain, its classification, taxonomy, relations (including all important hierarchies and constraints), and domain axioms. The ontology becomes a fundamental knowledge base that all other semantic agents should rely on and refer to in processing the information in the domain. Thus, it is very important to achieve a set of specified goals of accuracy and completeness in a specified context of use. Often such ontology is built from analysis of an information corpus and from the extraction of the relevant concepts and relations from that corpus. To incrementally build such ontology from unstructured text using semi-automated means, it is very important to present the knowledge workers or ontology construction agents with a ranked list of Subject-Predicate-Object (SPO) triples found in natural language text based on their semantic similarity to the specific search goals. The work in this thesis addresses such an ontology building effort in a specific domain: in our case it is the Cyber Security domain, with the methodology applicable to any general domains. The main problem addressed by this thesis is to develop a suitable methodology in order to get a highly accurate ranking of relevance for SPOs with respect to the search goals a knowledge worker or agent in the Cyber Security domain or any general domain might use.

Due to the rapid publishing of knowledge in unstructured texts on the World Wide Web, the need for efficient, high quality partitioning of texts into previously unseen categories is a major topic for applications. Ontology supports the shared understanding of the domain of interest by eliminating conceptual and terminological confusion among members of an online community. Concept and relation acquisition is an important aspect of ontology learning. For building a mutually agreeable ontology, it is very important to understand the context in which words are being used in unstructured natural language text. Context is any information that can be used to characterize the situation of an entity. In our work, an entity can be a subject, predicate, or object in a given SPO list. The semantics of words in the SPO is tied to a specific context. So, the primary objective of this research is to generate an accurate SPO ranking of relevance in a given context, i.e., search goals of the knowledge worker or semantic agent to build more accurate domain ontology.

Semantic similarity and semantic relatedness are often used interchangeably; however these two terms are not identical. Semantic similarity focuses on common points in the concept definitions, while semantic relatedness also takes into account the functional relations between the concepts. The relatedness measures may use a combination of the relationships that exist between two words, depending on the context or their importance. Semantic relatedness expresses the degree to which words are associated via any type (such as synonymy, meronymy, hyponymy, hypernymy, functional, associative and other types) of semantic relationships; while semantic similarity takes into consideration only hyponymy/ hypernymy relations. Therefore, we consider semantic similarity a special case of relatedness. A human can easily judge if a pair of words are related to each other in some way. For example, humans typically consider “attack” and “cyber attack” to be more closely related than “cyber attack” and “food.” Budanitsky and Hirst [18] consider semantic similarity measures to be appropriate only when similar entities such as “computer” and “mouse” or “apple” and “orange” are the subject of comparison. Often, the terms under study are related to each other with “is-a” hierarchy. For example, “cyber attack” is a kind of “attack.” So, it is a hyponym of attack. However, dissimilar terms may be also related semantically. For example “attack agent” affects “victim,” “attack agent” uses “technological aspect,” “means of attack” leads to “consequence.” In this case the two entities are semantically not similar, but are related by some relationship. Thus two entities are semantically related if they are semantically similar (close together in is-a hierarchy) or share with some other classical or non-classical relationships.

Computing semantic relatedness of natural language texts requires access to vast amounts of common-sense and domain-specific knowledge. A common problem is to get a good estimate of word usage in a particular context. Several methods exist to obtain such an estimate. However the accuracy of their results is not consistent. The primary challenge of this research is to carefully choose and integrate available methods for ranking the relevance of the subject-predicate-object (SPO) triples. A bad choice of context for word usage often leads to longer development times and poor quality of the ontology being constructed. Getting a good relevance score between SPOs according to the search goals of a knowledge worker is indeed difficult but imperative for creating high quality ontology.

We have developed our model in such a way that it can be used in different domains. In this thesis research, we use the Cyber Security domain as our application domain, as this domain is our KEWI[1] research group's major part of on-going projects. Cyber Security is a relatively new domain and has a large collection of domain specific terms. One way to determine the semantic relevance between the search goals of a knowledge worker and a given SPO's terms is to use a generic knowledge base such as the WordNet [13]. But since WordNet is a general purpose ontology, it may not contain many Cyber Security domain terms and concepts. Covering a maximum amount of terminology and terms in this domain was another challenging part for this thesis. As an outcome of this research, we maintained a domain specific corpus for storing the relevant terms and their relationships which may not be included in WordNet for building our SPO relevance measurements.

The main objective of this thesis thus, is to develop a methodology for ranking the relevance of SPO triples for a given concept, i.e. based on semantic similarity and relatedness to the search goals of a knowledge worker.

This thesis is organized as follows: section two discusses semantic similarity and semantic relatedness with examples. Section three presents an overview of related work. Our proposed method is described in section four, which contains a running example. Experimental results and evaluations are discussed in section five. Finally, the conclusion, our contributions and future work are outlined in section six.

[1] Knowledge Engineering and Web Intelligence

2 Semantic Similarity and Semantic Relatedness

Many literatures considered Semantic similarity and semantic relatedness to have the same meaning, which is not exactly true. Semantic relatedness is a more abstract version of a conceptual relationship, while semantic similarity is more specific case of semantic relatedness. In semantic relatedness, concept relations include "is-a-way-doing," "has-part," "is-a-cause," "is-a," etc. A measure of semantic similarity takes as input two concepts, and returns a numeric score that quantifies the similarity. Such measure represents an "is-a" relationship which resides in the taxonomy or ontology. For example, attack victim "is-a" government or user or country or business, etc. Much Ontology also includes additional relation between concepts. For example, the ontology built by Sousan et al in [1], has different relations like cyber attack that is related with agent, motive, coordination, etc., and could cause damage to systems, businesses, and websites. We believe that work in this thesis definitely provides a better way to select more relevant terms for a concept so as to build a more accurate ontology.

A number of semantic similarity methods have been developed in the past and different methods have been used for different purposes. Many of the methods are dependent on a general purpose ontology known as WordNet. WordNet is the product of a research project at Princeton University which has attempted to model the lexical knowledge of a native English speaker [13]. The system has the power of both an on-line thesaurus and an on-line dictionary, and much more. There is a multilingual WordNet for European languages which is structured in the same way as the English language

WordNet. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

WordNet was designed to establish the connections between four types of Parts of Speech (POS) - noun, verb, adjective, and adverb. The smallest unit in a WordNet is synset, which represents a specific meaning of a word. It includes the word, its explanation, and its synonyms. The specific meaning of one word under one type of POS is called a sense. Each sense of a word is in a different synset. Synsets are equivalent to senses "=" structures containing sets of terms with synonymous meanings. Each synset has a corresponding gloss, a term that defines the concept it represents. For example, the words night, nighttime, and dark constitute a single synset that has the following gloss: the time after sunset and before sunrise while it is dark outside. Synsets are connected to one another through explicit semantic relations. Some of these relations (hypernym, hyponym for nouns, and hypernym and troponym for verbs) constitute is-a-kind-of (holonymy) and is-a-part-of (meronymy for nouns) hierarchies.

2.1 Measures of Semantic Similarity

There are a number of measures that were developed to quantify the degree to which two words are semantically related using information drawn from semantic networks. These measures can be generally categorized into two main kinds metrics:

knowledge based measure, and corpus based measure [14]. Both measures have different methods. Figure 1 shows some of the important approaches for both measures.

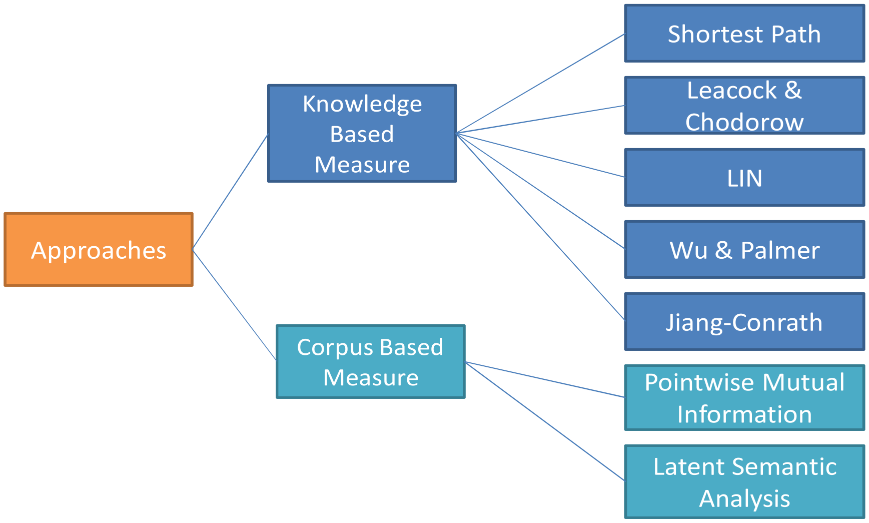


Figure 1: Different approaches for measure Semantic Similarity between words

2.1.1 Knowledge-based Measures

Knowledge based measures [16] identify the semantic similarity between two words by calculating the degree of relatedness among words using information from a dictionary or thesaurus. It makes use of the relations and the hierarchy of a thesaurus, which is generally a hand-crafted lexical database such as WordNet. For example the Leacock & Chodorow method [16] counted the number of nodes of the shortest path between two concepts. The work by Resnik [17] also used WordNet to calculate the semantic similarity.

The **shortest path** similarity is determined as:

$$Sim_{path} = \frac{1}{length}$$

Equation 1 Shortest path

Where *length* is the length of the shortest path between two concepts using a node-counting (including the end nodes) approach according to their relational positions in a graph structure of the concepts in the WordNet.

The **Leacock & Chodorow** [16] similarity is determined as:

$$Sim_{lch} = - \; log \frac{length}{2 \times D}$$

Equation 2 Leacock & Chodorow

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

The **Lesk** similarity [16] of two concepts is defined as a function of the overlap between the corresponding definitions, as provided by a dictionary. It is based on an algorithm proposed by Lesk as a solution for word sense disambiguation.

The measure introduced by Resnik [7] returns the information content (IC) of the the least common subsumer (LCS) of two concepts denoted as:

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

Equation 3 Resnik Method

Where IC is defined as:

$$IC\ (C) = -\ \log P(c)$$

Equation 4 Information Content

and P(c) is the probability of encountering an instance of a concept c in a large corpus.

Jiang & Conrath [16] introduced another approach to measure similarity:

$$Sim_{jnc} = \frac{1}{IC(concept_1) + IC\ (concept_2) - 2 \times IC(LCS)}$$

Equation 5 Jiang & Conrath

The **Lin measure** [17] of semantic relatedness of concepts is based on his Similarity Theorem. It states that the similarity of two concepts is measured by the ratio of the amount of information needed to state the commonality of the two concepts to the amount of information needed to describe them. The commonality of two concepts is captured by the information content of their LCS and the information content of the two concepts themselves. This measure turns out to be a close cousin of the Jiang–Conrath measure, although they were developed independently:

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

Equation 6 Lin Measure

The **Wu & Palmer** similarity [16] metric measures the depth of two given concepts in the WordNet taxonomy, and the depth of the LCS, and combines these figures into a similarity score:

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

Equation 7 Wu & Palmer measure

Here are some examples (Fig. 2) to explain how to use this formula, where hyponym taxonomy in WordNet is used for path length similarity measurement [20].

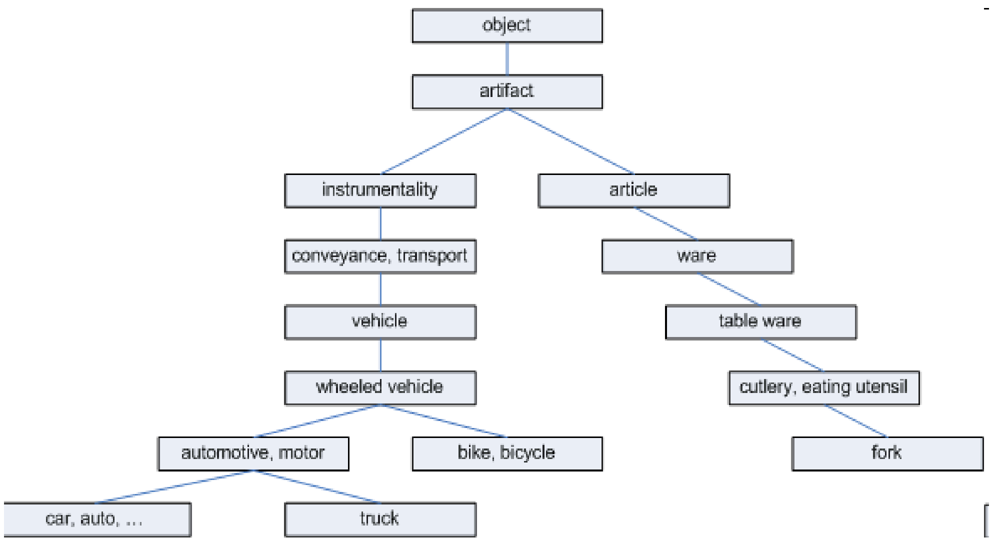


Figure 2 Example of Lexical Database

In the figure 2, we see that the length between car and auto is 1, car and truck is 3, car and bicycle is 4, car and fork is 12. A shared parent of two synsets is known as a subsumer. The least common subsumer (LCS) of two synsets is the sumer that does not have any children as shown in figure 3. In other words, the LCS of two synsets is the most specific subsumer of the two synsets. The LCS of {car, auto..} and {truck..} is {automotive, motor vehicle}, since the {automotive, motor vehicle} is more specific than the common subsumer {wheeled vehicle}. The depth of LCS for car and truck is 7 based on the figure 2.

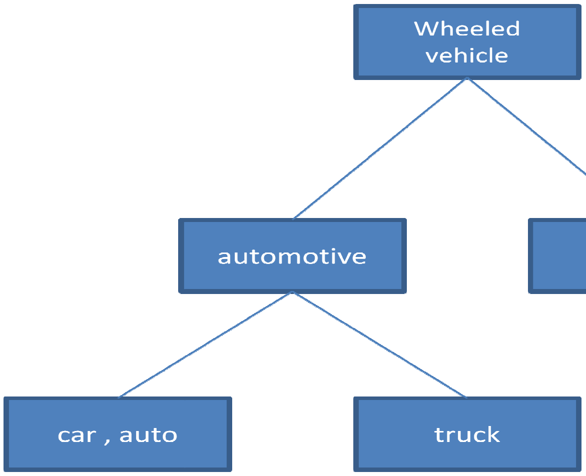


Figure 3 Least Common Subsumer (LCS)

$$\mathtt{Sim(car,truck)}_{\text{wup}} = (2*7)/(8+8) = 0.875$$

$$\mathtt{Sim(car,truck)}_{\text{wup}} = (2*8)/(8+8) = 1$$

$$\mathtt{Sim(car,truck)}_{\text{wup}} = (2*6)/(8+7) = 0.8$$

2.1.2 Corpus Based Measure

Corpus-based measures of word semantic similarity tries to identify the degree of similarity between words using information exclusively derived from large corpora. There are two types of methods under corpus based measures: **Pointwise mutual information (PMI)** [9] **and Latent Semantic Analysis (LSA)** [14]. Both of these approaches are statistical.

The Pointwise Mutual Information (PMI) between two words w_1 and w_2 captures how likely it is to find B in a text given that we know that the text contains A. It is a co-occurrence metric, in that it normalizes the probability of co-occurrence of the two words with their individual probabilities of co-occurrence. Thus PMI method is based on term co-occurrences processed using frequency counts over large corpus. Given two words w_1 and w_2, their PMI is measured as:

$$\text{PMI}\ (w_{1,}\ w_2) = \log_2 y(p(w_{1,}\ w_2)/\ p(w_1)^*\ p(w_2))$$

Equation 8 Pointwise Mutual Information Measure

Where P (w_1 & w_2) is the probability w_1 and w_2 co-occur in the same document (means within a given word window size). P (w) is the probability that a word occurs in the document. The similarity between words w_1 and w_2 is then estimated by their PMI score. PMI measures the degree of statistical dependence between the words.

Another popular approach is the Latent Semantic Analysis (LSA) where the term co-occurrences are captured by means of dimensionality reduction operated by singular value decomposition (SDV). **LSA** attempts to solve the problem of how to find the relevant documents from search words based on meanings or concepts behind the words. It constructs a matrix [A] from given text, in which the row vectors represent words and the column vectors represent chunks of text. The method then calculates the weight of each cell by using tf-idf score. Apply Singular Value Decomposition (SVD) to [A] to decompose into a product of three matrices. SVD reduces a matrix to a given number of dimensions. This may convert a word level space into a semantic or conceptual space. The similarity of two words is measured by the cosine of the angle between their corresponding compressed row vectors.

The tf-idf weight [24] (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search as a central tool in scoring and ranking a document's relevance given a user query.

Mathematical details

The term count in the given document is simply the number of times a given term appears in that document [24]. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j. Thus we have the term frequency, defined as follows.

$$\mathrm{tf}_{\mathrm{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ is the number of occurrences of the considered term (t_i) in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j, that is, the size of the document $| d_j |$.

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

with

- $| D |$: Cardinality of D, or the total number of documents in the corpus
- $|\{d : t_i \in d\}|$: number of documents where the term t_i appears (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use $1 + |\{d : t_i \in d\}|$

Then

$$(\mathrm{tf\text{-}idf})_{\mathrm{i,j}} = \mathrm{tf_{i,j}} \times \mathbf{idf_i}$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. The tf-idf value for a term will be greater than zero if and only if the ratio inside the idf's log function is greater than 1. Depending on whether a 1 is added to the denominator, a term in all documents will have either a zero or negative idf, and if the 1 is added to the denominator a term that occurs in all but one document will have an idf equal to zero.

For example, consider a document containing 100 words wherein the word cow appears 3 times. Following the previously defined formulas, the term frequency (TF) for cow is then (3 / 100) = 0.03. Now, assume we have 10 million documents and cow appears in one thousand of these. Then, the inverse document frequency is calculated as log (10 000 000 / 1 000) = 4. The tf-idf score is the product of these quantities: 0.03 × 4 = 0.12.

As we saw from the above descriptions, there are many methods available for measuring semantic similarity between word pairs. Some use a knowledge base approach which is based on lexical database like WordNet for different methods, as to Resnik, Jiang & Conrath, lin, Leacock & Chodorow e.t.c. Others use corpus based methods, which use statistical approaches like the PMI method or LSA method. Some use a hybrid approach which is a combination of both knowledge based measure and corpus based measure. Statistical approaches are not very efficient because of lack of relevant data and terms in a particular domain. Hybrid approaches attempt to address this problem by using both the approaches of lexical database for general purpose words and local corpus for domain specific data. We use the hybrid approach in our thesis work to cover both general purpose words as well as any domain specific words, to make our results more close to human perception.

We conduct the algorithm for determining the semantic similarity as implemented in the WordNet Similarity Package [13]. We analyzed the results of different methods on a selected dataset which we obtained from Yuhua Li's paper[10]. In this paper, the authors collected human ratings for the similarity of pairs of sentences following existing designs for word similarity measures. The participants consisted of 32 volunteers, all native speakers of English educated to graduate level or above. They compared the result of their methods to three other results: (1) Human Similarity, (2) method of [10], and (3) Text Similarity of ISLAM [9]. Table 2 shows a comparison between three results along with our experimental results, which we obtained from WordNet package, in order to compare the methods. All of these measures assume as input a pair of words, and return a value indicating their semantic relatedness.

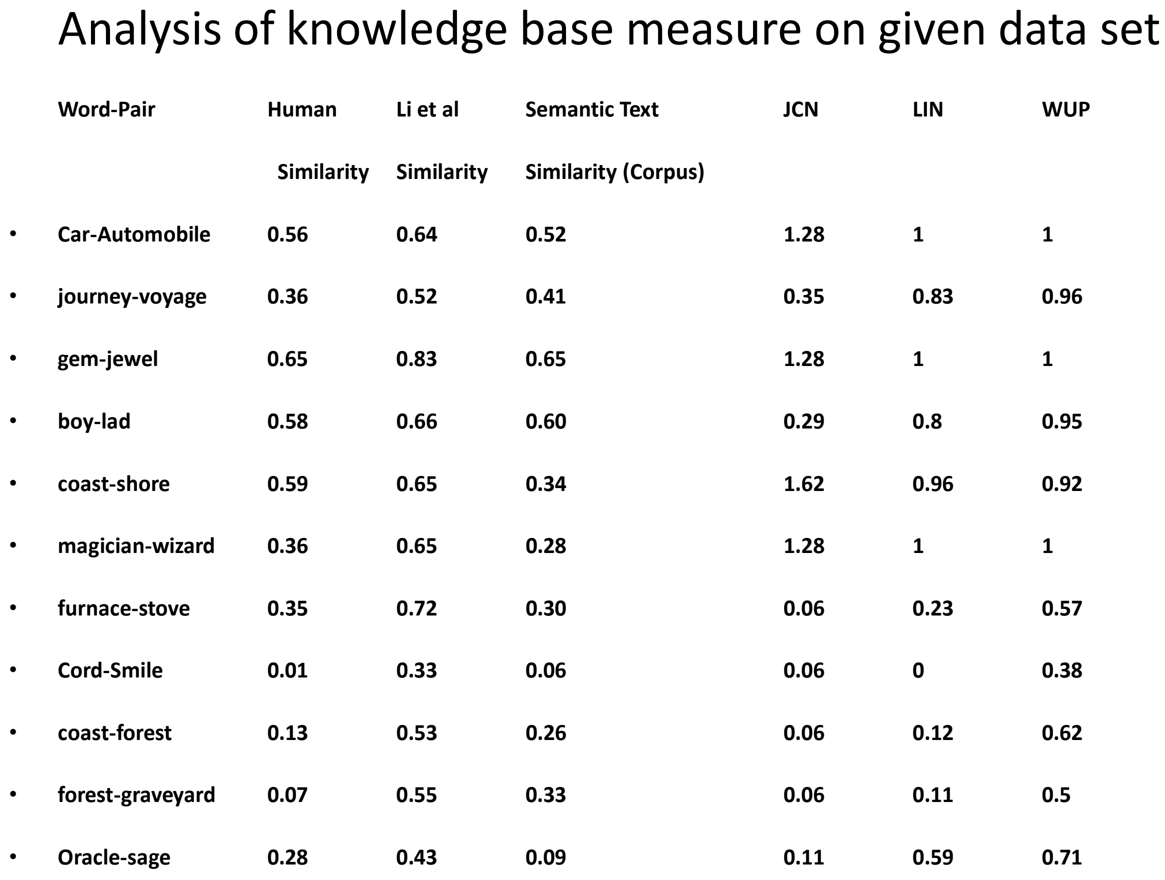
Method	**Formula**

Jiang & Conrath (JCN)	$\frac{1}{IC(concept_1) + IC\ (concept_2) - 2 \times IC(LCS)}$
	IC = information content
	LCS = Least Common Subsumer

LIN	$\frac{2 \times IC\ (LCS)}{IC(concept_1) + IC\ (concept_2)}$ Concept = Given Word

Wu and Palmer (WUP)	$\frac{2\ \times depth(LCS)}{depth\ (concept_1)\ +\ depth\ (concept_2)}$ Depth = Height of the word in Lexical database

Table 1 Formulas used for comparison (JCN, LIN, WUP)



Analysis of knowledge base measure on given data set

Word-Pair	Human	Li et al	Semantic Text	JCN	LIN	WUP
	Similarity	Similarity	Similarity (Corpus)			
• Car-Automobile	0.56	0.64	0.52	1.28	1	1
• journey-voyage	0.36	0.52	0.41	0.35	0.83	0.96
• gem-jewel	0.65	0.83	0.65	1.28	1	1
• boy-lad	0.58	0.66	0.60	0.29	0.8	0.95
• coast-shore	0.59	0.65	0.34	1.62	0.96	0.92
• magician-wizard	0.36	0.65	0.28	1.28	1	1
• furnace-stove	0.35	0.72	0.30	0.06	0.23	0.57
• Cord-Smile	0.01	0.33	0.06	0.06	0	0.38
• coast-forest	0.13	0.53	0.26	0.06	0.12	0.62
• forest-graveyard	0.07	0.55	0.33	0.06	0.11	0.5
• Oracle-sage	0.28	0.43	0.09	0.11	0.59	0.71

Table 2 Analysis of Knowledge base measures

Based on the analysis in table 2, we found that Lin's [17], and Wu & Palmer's [4] are the most appropriate methods for our purposes for calculating semantic similarity because we were looking for methods which could return the result in 0 to 1 normalized form. We also looked at methods that should have output of results close to the other considered results and found that these two methods were appropriate for our case. The JCN method was not considered because its output was not in 0 to 1 normalized range. We provide below a description of each of these two methods. So, overall we concentrated on only two types of knowledge based measures and one type of corpus based measure.

3 Related work

Evaluating semantic relatedness and finding semantic similarity between documents, sentences or words is a problem with a long history. As mentioned in the introduction, our work is to compare a given concept with a list of SPOs and rank them based on their relevance of semantic similarity in order to help a semantic agent or knowledge worker to build a reliable and authentic ontology or find the right reason for cause and effect or extract the correct pattern from natural language process. The system can help to find the important term in term extraction pool for a given concept in order to construct ontology in order to reduce the work load of the knowledge worker. Research related to measuring similarity between sentences and documents in English are extensive [8, 9, 10], but there has been very little work which relates to semantic similarity between SPOs. Most of the sentence similarity measures mainly concern 'calculating' the availability or non-availability of words in the compared sentences [9]. Therefore, the word overlap measures [25], tf-idf measures [25], relative frequency measures [25] and probabilistic models [25] have been the popular methods for evaluating the similarity. Some of the research used the word co-occurrence methods which are known as "bag of words" method to find the similarity between two sentences. But, this kind of methods generally use in Information Retrieval model [26]. For calculating the semantic similarity between two sentences, some researchers simply aggregate the similarity values of all the word pairs [27]. But in our case, we derived the semantic similarity vector and then use cosine similarity formulas for finding overall semantic similarity between two SPOs. We also considered the important of the position of the words in a given SPO. We work on SPO in order to reduce the noise and give more important to the main words in a sentence.

Many techniques have been proposed for evaluating semantic similarity between words in hierarchies such as WordNet and GeneOntology. The approaches can be classified into two categories: edge based and node based approaches. The edge based approach is the simplest similarity measure, and computes the distance between two concepts based on the number of edges found on the path between them [7]. In the node based approach, Lin [6] defined the similarity between two concepts as the ratio between the amount of information needed to state the commonality between these two concepts and the information needed to fully describe them. Wu [4] found the path length to the root node from the least common subsumer (LCS) of the two entities, which was the most specific entity they share as an ancestor. The value is scaled by the sum of the path lengths from the individual entities to the root. Leacock [3] found the shortest path between two entities, and scaled that value by the maximum path length in an “is–a” hierarchy in which they occurred. Recently, new work by Vincent D. Blondel [5] defines more sophisticated topological similarity measures, based on graph matching from discrete mathematics. These new graph-based measures suit the particularities of the new ontologies built with more expressive languages like OWL [12]. However, these methods are not applicable for ontologies with different types of relationships. We have selected two methods using WordNet based on our analysis on different datasets.

Some recent research efforts have focused on using Wikipedia to improve coverage with respect to traditional thesauri-based methods. Nowadays, Wikipedia is rapidly growing in size, and it is not difficult to find new terms and named entities on it. But Wikipedia doesn't have many words, especially in specific domains like the cyber security domain. So, along with WordNet and Wikipedia, we also considered the local corpus for covering all the terminologies and terms in a particular domain. In our case, we considered the local corpus in the cyber security domain.

Many researchers have done good work to calculate the semantic similarity and relatedness in general purpose ontology like WordNet, which is not from any specific domain. In this proposed method, we followed the [9] [10] approach with some modifications. Our work mainly concentrated on improving the performance of the measurements in a specific domain which can help to detect or extract the more relevant term to insert into the ontology of that specific domain. For example, in this paper we have concentrated on applying our method to improve the ontology building in the cyber security domain. We used three different kinds of resources: WordNet, Wikipedia, and local corpus. WordNet and Wikipedia are used to cover all the general purpose words and terms, while our local resource is based on a specific domain, i.e., cyber security domain to cover the domain specific words and terms. Our local corpus mainly contains the news articles in text files. This relevance measurement model can be used for any domain for comparing the given concept with the SPO list. For using this model for any specific domain, we just need to change the local corpus.

For using Wikipedia, we use the DISCO (extracting DIStributionally related words using CO-occurrences) system [15]. DISCO is a Java class which allows us to use the API for measuring the semantic similarity between arbitrary words. The similarities are based on the statistical analysis of very large text collections. We used the Wikipedia-2008 English version, which contain 220,000 words and has a corpus size of 267 million tokens.

4 The Proposed Method

Computing semantic similarity between an SPO and a given concept is an important function in multi ontological applications such as semantic data integration and ontology mapping. This thesis work has concentrated on the ranking of relevance of SPO triples in order to create a good quality of ontology. The work used existing tools [1] to generate the SPO triples for a given set of text corpus. After generating the triples, we have compared each sequence of the extracted triples with the present seed ontology or new concept to create a list of relevant SPO triples to allow the user to select the concept relevant terms. This comparison was more critical in order to build ontology because a wrong selection of the term can lead to poor quality of results.

Semantic similarity is a confidence score that reflects the semantic relation between the meanings of two SPOs. For a given concept and a single SPO, semantic similarity measures how similar the meaning of the given concept and the SPO is. The higher the score, the more similar the meaning of the SPO is for the concept. As we know, semantic similarity represents the lexical similarity [10]. Word order similarity provides syntactic information about the relationship between words: which words appear in the sentence and which words come before or after other words. Both semantic and syntactic information (in terms of word order) play a role in conveying the meaning of the SPO triples.

The proposed method derives text similarity from semantic and syntactic information contained in the compared texts. A text is considered to be a sequence of words each of which carries useful information. The words, along with their combination structure, make a text convey a specific meaning. We considered SPO as a text in this paper which carries the semantic and syntactic information.

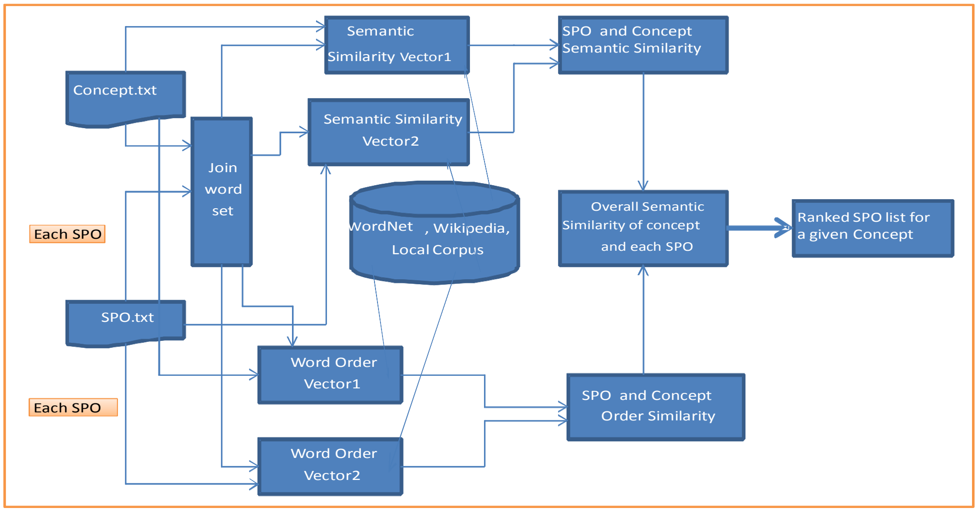

Concept.txt
Join word set
Each SPO
SPO.txt
Each SPO
Semantic Similarity Vector1
Semantic Similarity Vector2
WordNet , Wikipedia, Local Corpus
Word Order Vector1
Word Order Vector2
SPO and Concept Semantic Similarity
Overall Semantic Similarity of concept and each SPO
SPO and Concept Order Similarity
Ranked SPO list for a given Concept

Figure 4 Block Diagram of Proposed Method

Figure 4 shows the block diagram of the proposed method for computing and ranking the relevance of a set of SPOs for a given concept based on their semantic similarity or semantic relatedness in order to select a more appropriate term for a given concept to build a reliable and authenticate ontology. As we know, many previous methods used only a fixed set of vocabulary for calculating the similarity. Our method dynamically creates a joint word set from a SPO and a given concept, which contains only unique words present in both concept and SPO, in order to create a semantic vector of the same dimension for both the SPO and the concept individually. Next, we derived semantic similarity vectors for a concept and an SPO with the help of three resources. First is the lexical database WordNet, second is Wikipedia, and the third one is our local corpus. We also created the word order similarity vectors for calculating the syntactic information for a given concept and SPO using the same resources. Since each word in

SPO and Concept contributes in different way to the meaning of the whole combination of subject -> predicate -> object. The importance of a word is weighted with respect to the semantic similarity value by using WordNet, Wikipedia, and local corpus. Based on semantic similarity and word order similarity, we calculated the overall similarity between the given concept and each SPO and then ranked them based on overall similarity value.

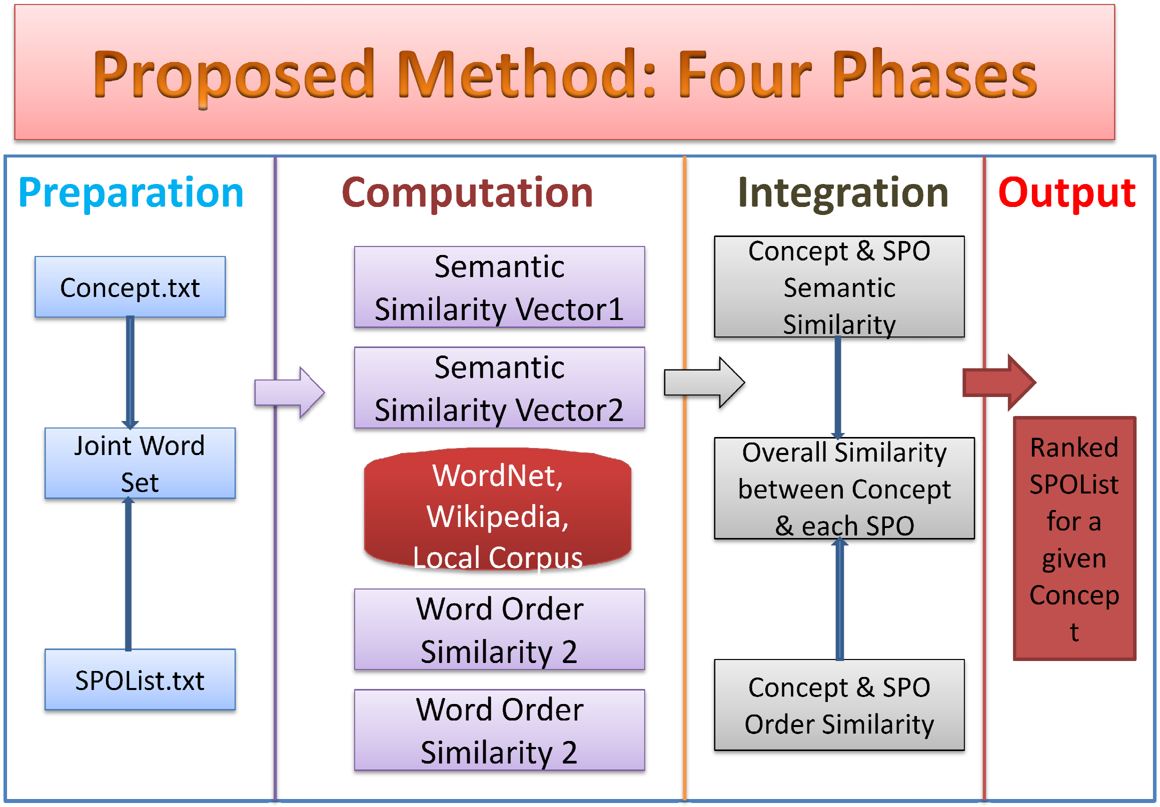
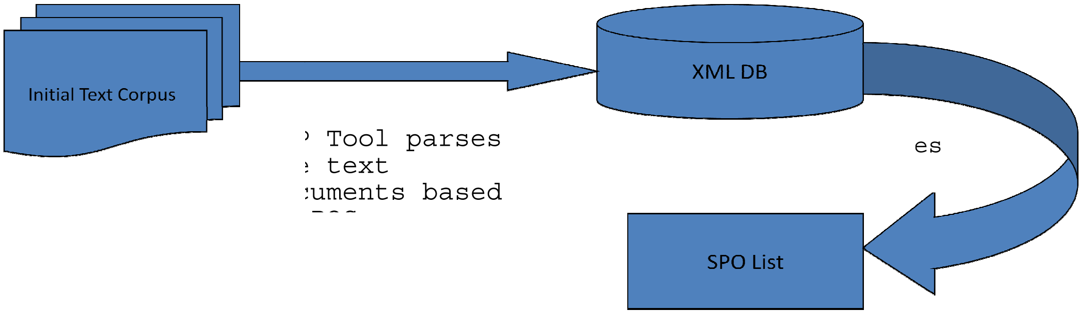

Proposed Method: Four Phases
Preparation
Computation
Integration
Output
Concept.txt
Joint Word Set
SPOList.txt
Semantic Similarity Vector1
Semantic Similarity Vector2
WordNet, Wikipedia, Local Corpus
Word Order Similarity 2
Word Order Similarity 2
Concept & SPO Semantic Similarity
Overall Similarity between Concept & each SPO
Concept & SPO Order Similarity
Ranked SPOList for a given Concep t

Figure 5 Proposed Method in four different phases

Figure 5 represents the proposed method in four different phases. The first phase is the preparation phase, the second is the computation phase, the third is the integrated phase, and the fourth is the output phase. In the first phase, we prepared the joint word set from the concept and each SPO which contain unique words. In the computation phase, we derived semantic similarity vector and word order similarity vector with the help of three resources: WordNet, Wikipedia, and local corpus. The next phase is the integration phase where we combined the results received from semantic similarity and word order similarity in order to get the overall similarity between the concept and each SPO. The last phase is output phase, where we ranked the set of SPOs for a given concept based on their relevance of similarity.

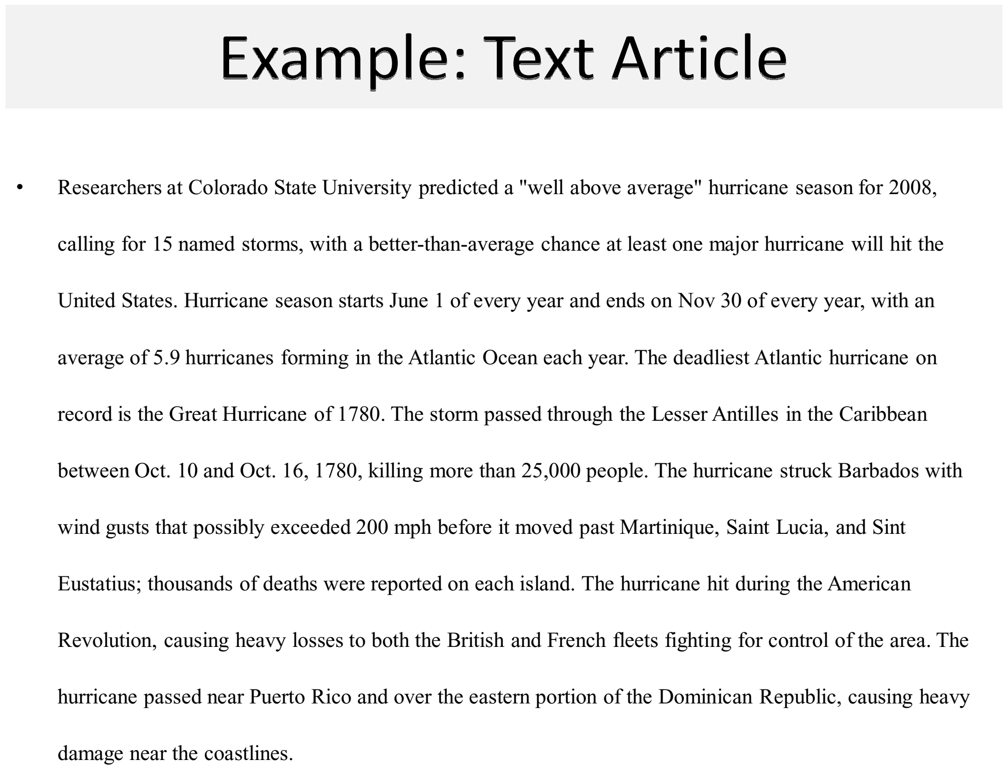
In the first phase, inputs are concept and a set of SPOs which is in the form of an SPO (subject -> predicate -> object). For generating the SPOs list from a natural language text, like a news article or text article, we used NLP tool [1] developed by Dr. Sousan. Figure 6 is the block diagram for extracting the SPO from a natural language text.



Used tool developed by ***Dr. William Sousan*** *[1]*

Figure 6 Diagram for creating SPO List from Unstructured data

The NLP tool parses the text article and creates an XML database based on their part of speech. Again this tool parses the XML DB and generates SPO lists, which we use in our thesis work as an input. Figure 7 represent an example of a text article.

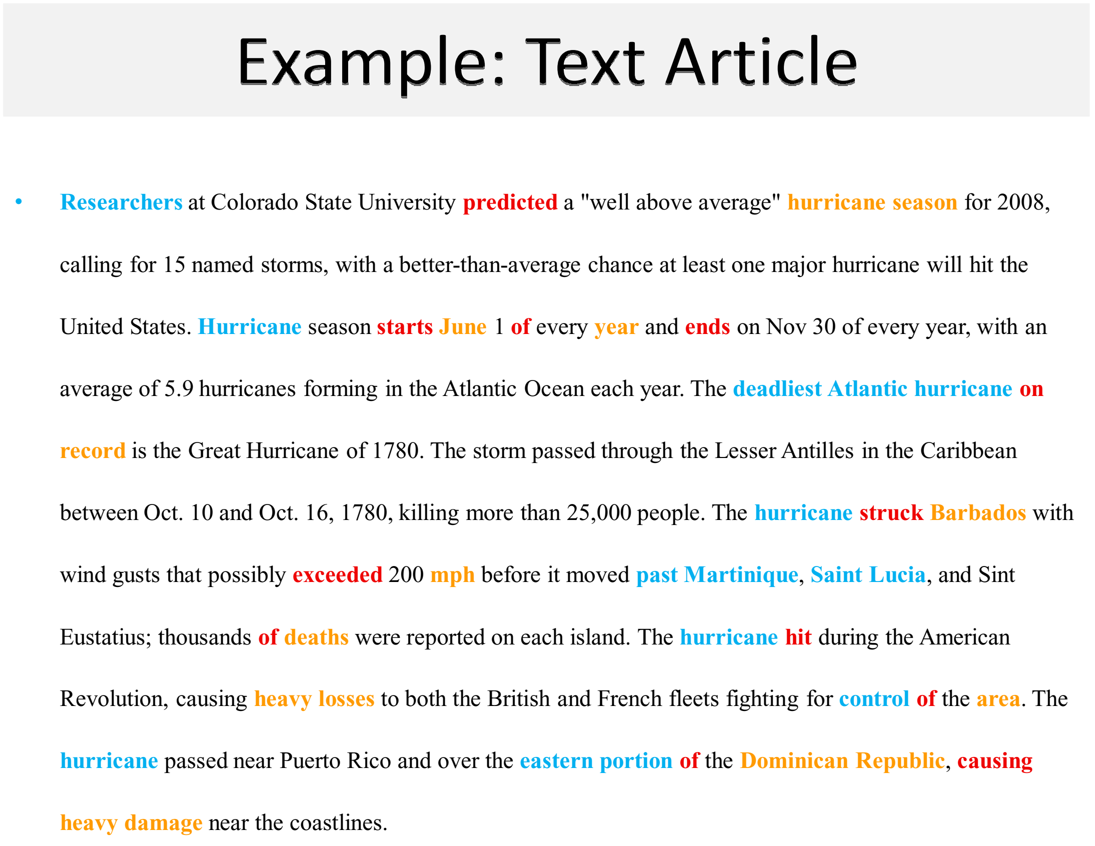


Example: Text Article

- Researchers at Colorado State University predicted a "well above average" hurricane season for 2008, calling for 15 named storms, with a better-than-average chance at least one major hurricane will hit the United States. Hurricane season starts June 1 of every year and ends on Nov 30 of every year, with an average of 5.9 hurricanes forming in the Atlantic Ocean each year. The deadliest Atlantic hurricane on record is the Great Hurricane of 1780. The storm passed through the Lesser Antilles in the Caribbean between Oct. 10 and Oct. 16, 1780, killing more than 25,000 people. The hurricane struck Barbados with wind gusts that possibly exceeded 200 mph before it moved past Martinique, Saint Lucia, and Sint Eustatius; thousands of deaths were reported on each island. The hurricane hit during the American Revolution, causing heavy losses to both the British and French fleets fighting for control of the area. The hurricane passed near Puerto Rico and over the eastern portion of the Dominican Republic, causing heavy damage near the coastlines.

Figure 7 Example of unstructured data (news article)

Figure 8 represents the subject, predicate and object in different colors in a text article.

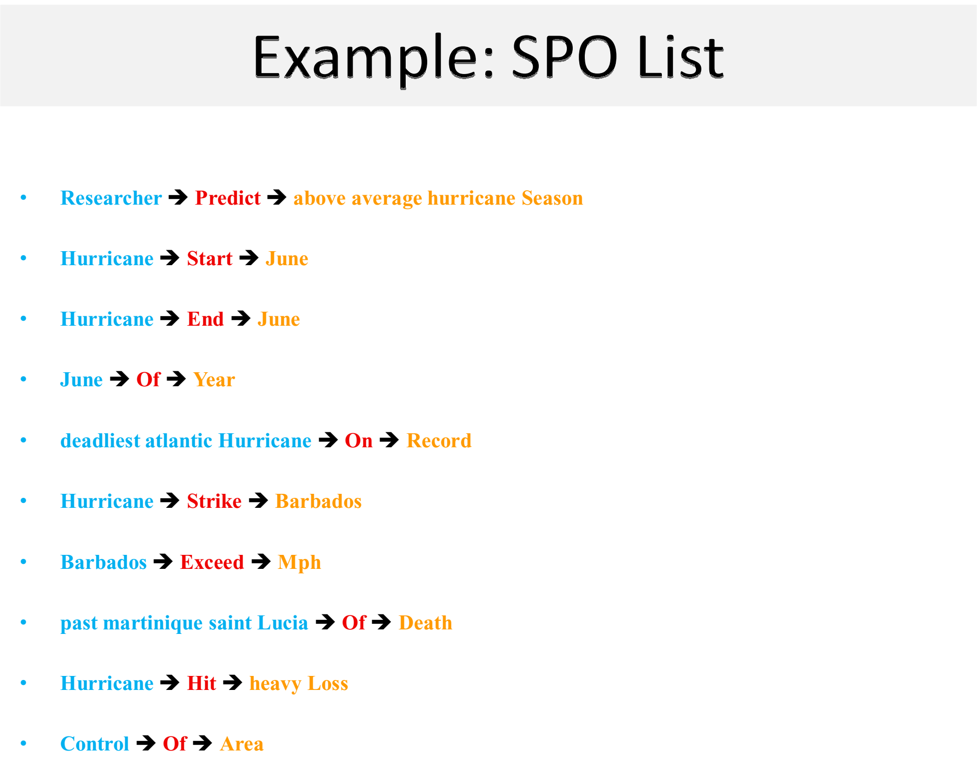


Example: Text Article

- **Researchers** at Colorado State University **predicted** a "well above average" **hurricane season** for 2008, calling for 15 named storms, with a better-than-average chance at least one major hurricane will hit the United States. **Hurricane** season **starts June** 1 **of** every **year** and **ends** on Nov 30 of every year, with an average of 5.9 hurricanes forming in the Atlantic Ocean each year. The **deadliest Atlantic hurricane on record** is the Great Hurricane of 1780. The storm passed through the Lesser Antilles in the Caribbean between Oct. 10 and Oct. 16, 1780, killing more than 25,000 people. The **hurricane struck Barbados** with wind gusts that possibly **exceeded** 200 **mph** before it moved **past Martinique**, **Saint Lucia**, and Sint Eustatius; thousands **of deaths** were reported on each island. The **hurricane hit** during the American Revolution, causing **heavy losses** to both the British and French fleets fighting for **control of** the **area**. The **hurricane** passed near Puerto Rico and over the **eastern portion of** the **Dominican Republic**, **causing heavy damage** near the coastlines.

Figure 8 Represent the SPO in a text article

Figure 9 shows the SPO list which we got after parsing the xml file by using the NLP tool



Example: SPO List

- Researcher → Predict → above average hurricane Season
- Hurricane → Start → June
- Hurricane → End → June
- June → Of → Year
- deadliest atlantic Hurricane → On → Record
- Hurricane → Strike → Barbados
- Barbados → Exceed → Mph
- past martinique saint Lucia → Of → Death
- Hurricane → Hit → heavy Loss
- Control → Of → Area

Figure 9 Example of SPO List generated by the NLP tool

Before going ahead, we want to discuss some of the important concepts which we used in our proposed method. Specifically, these three concepts are key:

1. Lexical semantic vector
2. Second order co-occurrence PMI Method for using local corpus
3. Semantic similarity calculation based on word order

4.1 Lexical Semantic Vector

Lexical semantics is a branch of semantics (the study of language meaning) that studies the meanings and relations of words. In another way, lexical semantics is the study of word meaning. So, it is a subfield of linguistic semantics, which studies how and what the words of a language denote. Linguistics is the scientific study of human language. It encompasses a number of sub-fields. An important topical division is between the study of language structure (grammar) and the study of meaning (semantics and pragmatics). Thus, lexical semantics covers the theories of the classification and decomposition of word meaning, and the relationship of word meaning to sentence meaning and syntax.

For creating a lexical semantic vector, let's consider two SPOs, say sp_1 and sp_2. Denote the joint word set of these two SPOs as $sp = sp_1 \cup sp_2$, which contain all the unique words from both SPOs. The joint word set, sp, can be viewed as a representation of the semantic information for the composite SPOs. The vector derived from the joint word set is called the lexical semantic vector, denoted as $\check{s}$. Each entry of the semantic vector corresponds to a word in the joint word set, so the dimension of the vector is equal to the number of words in the joint word set. The value for an element of the lexical semantic vector is determined by the semantic similarity of the corresponding word in the SPO pair or the concept and the joint word set. The semantic similarity between two SPOs is defined as the cosine coefficient between the two vectors.

$$\boldsymbol{S_s} = \frac{\breve{\boldsymbol{S_1}} \,.\, \breve{\boldsymbol{S_2}}}{\|\breve{\boldsymbol{S_1}}\| \,.\, \|\breve{\boldsymbol{S_2}}\|}$$

Equation 9 Cosine Similarity from two lexical semantic vectors

Where $\check{s}_1$ and $\check{s}_2$ denote the lexical vectors derived from the concept and the SPO for calculating the semantic similarity.

In the following section, we will discuss each of the above computational steps. Since semantic similarity between words is used both in measuring concept and SPO semantic similarity and word order similarity, we will first describe the method for measuring word semantic similarity.

4.2 Second Order Co-occurrence - PMI Method for Relevance Measurement with Local Corpus

For using a local corpus [19] as one of the resources, we first created a single file of local corpus from 140 existing news articles collected in the cyber security domain. A Natural Language Processing (NLP) tool was used to process each file from the text corpus and extract the individual sentences from each file. Each sentence was then partitioned into a list of words, with a removal of the stop words. It is common to ignore the stop words that are frequently occurred in a language processing. The process also removes insignificant words such as those that appear in a database record, article, web page, etc. In our method, we used a popular stemming algorithm called Porter stemming. Porter stemming is a process of removing the common morphological and inflexional endings of words. The process of stemming converts each word in the search index to its basic root or stem (e.g. 'coming' to 'come') so that variations on a word ('comes', 'came', 'coming', 'come') are considered equivalent when searching. This generally provides more relevant search results.

After using the stemmer algorithm, we saved all the words in our local disk as a large corpus of text for future use. At present our local corpus consists of 522,731 words. The local corpus contains all the words in the same order as they appeared in the text file. Then, we created a local dictionary which is a set of all the unique word present in the local corpus. Initially, we consider all the unique words, and the total number of words was 6,565.Due to the large size of the dictionary, along with other coding style, total calculation time for the semantic measure for one pair of words took approximately 45sec to 50 sec, which was very inefficient. So, to speed this process, we considered only those words in the local dictionary which were present in our application domain but neither in WordNet nor in Wikipedia. The experimentation of this thesis only contains those words which are related to the cyber security domain. After the word reduction according to our application domain, the total words present in our local dictionary are 1,359.

Initially, to calculate the semantic measure for a pair of words, we were searching each word in the local corpus individually, so for two words, our program was browsing the local corpus of 522,731 words two times, which was also taking extra time. Then, we improved our coding style and now for two words, it refers to the local corpus only one time, thus saving extra time. After making these computational improvements, the efficiency of our program increased drastically, and now it takes only 5 to 6 sec to calculate the semantic similarity for a given pair of words.

The following steps are involved in calculating the semantic similarity from the local corpus:

1. Calculate word frequency $f^{t}(t_i)$ means how many times the word t_i is present in the entire local corpus. Where t_i represents the words present in the corpus and i = 1, 2, 3……n. Word frequency

2. Calculate how many times the given words appear together with each word in the dictionary in a window size of $2\alpha + 1$ words. Where α can be 1, 2, 4……n. In my case, I have taken 5

3. If the frequency with which the words appear together is greater than 0 then the Pointwise Mutual Information (PMI) value, denoted as f^{pmi}, is calculated for those words based on the equation 10.

$$f^{pmi}(t_i, W) = \log_2 \frac{f^{b}(t_i, W) \times m}{f^{t}(t_i) f^{t}(W)},$$

Equation 10 PMI for words appear together

t_i represents the each word present in the local corpus, W is word for which we calculated the PMI value, f^b(t, W) is the frequency with which the word W and t_i appear together, and m is the total number of words present in the local corpus.

4. Once all the possible PMI values for both of the words are calculated individually, they are then ranked in decreasing order of their PMI values. The PMI values are further processed to obtain a β value for each word. The β value depends on the word and word's frequency in the corpus as shown in equation 11.

$$\beta_i = \left(\log\ (f^t(W_i))\right)^2 \frac{(\log_2(n))}{\delta}, \text{ where } i = 1, 2$$

Equation 11 Bita value calculation

n is the total number of unique word present in the local dictionary. In our case, the value of n is 1359. n can vary based on the local dictionary size. The value of δ depends on the size of the corpus, if the size of the corpus is small, then we should select the smaller value of δ. In our case we have considered δ as 5.5. The value of δ is important because based on this; the value of β can change. If we lower the value of β by considering the value of δ high, then we lose some important / interesting words, and if we increase the value of β by considering the value of δ low, we consider more words common to both w_1 and $w_{2,}$ and this significantly degrades the result.

We calculate β_1 and β_2 values for word w_1 and word w_2 respectively. After calculating the beta value, we defined the set of words P and Q for each word w_1 and w_2

which store in descending order by their PMI value and take the top-most β words having $f^{pmi}(t_{i,}$ w) > 0. So, for word w_1, the set of P is defined as below

$$P = \{P_i\} \text{ where } i = 1, 2, 3\ldots, \beta_1$$

Similarly for the word w_2, the set of words Q define as below

$$Q = \{Q_i\} \text{ where } i = 1, 2, 3\ldots, \beta_2$$

Finally the semantic PMI similarity between two words w_1 and w_2 from the local corpus is calculated as below

$$sim\ (w_1\ ,w_2) = P/\beta_1 + P/\beta_2$$

4.3 Semantic Similarity calculation based on word Order

From the above discussion, we know that for calculating the overall semantic similarity between a given concept and SPO, the computation of syntactic similarity [10] is also very important, along with the lexical similarity computation. Syntactic study concerns the sequences in which words are put together to form sentences, clauses, phrases, and SPO. In English, the usual sequence for words is subject, verb, and object. For example "The boy loves his dog" follows standard subject-verb-object order, and switching the order of such a sentence would change the meaning or make the sentence meaningless. For example: the sentence “The dog loves his boy” has a different meaning from the above sentence. Computing similarity according to the word co-occurrences only and ignoring syntactic information may work well for long texts, especially if the long texts contain adequate information in terms of the co-occurrences for similarity measurement (i.e., they have a sufficient number of co-occurring words). However, for a similarity computation in general, and for a computation performed over a text without sufficiently large amount of information in the form of co–occurrence, the computation cannot be reliably carried out. In our thesis work, we are calculating the semantic similarity between two SPOs, which contain very few words, so ignoring the word order may become a significant source of divergence with the results expected by a knowledge worker.

4.3.1 Word order similarity between concept and SPO

Let us consider that we have one concept as C and one SPO as S for comparison which contain the **same number** of words except two words are swapped, as shown in figure 10.

C = boy -> loves -> his dog

S = Dog -> loves -> his boy

Figure 10 Example of two SPOs for word order similarity calculation

If we calculate the semantic similarity value based on the semantic similarity vectors, we get the semantic value one for the C and the S which means both the C and the S have the same meaning, which is actually not true. So, we also consider the importance of the order of the words in an S and calculate the word order similarity. By using the word order similarity method, we find that these C and S are not same in meaning.

Seeing figure 10, it is very easy for a human being to evaluate that the two SPOs have different meanings though they contain the same words. Yuhua Li [10] considered the word order information for calculating the semantic similarity between two small sentences

For example, the joint word set for concept C is listed below:

CS = {boy, loves, his, dog}

We assigned a unique number to each word in C. The numbers are assigned in the order in which the words appear in the S. We then assigned the word order to the words in the S. The number assigned to words in S is based on the following three rules:

1. If the same word is present in C, then assign the same number to this word in S

2. If the word is not present in C, then find if there is any word in C that matches closely with the given word S. If there is one such word present in C, assign the order number of that word in C for this word in S.
3. If the above two methods do not work i.e. if no same word is present in C and no word has a similar meaning to the selected word from S, then assign order number zero to this word.

Based on the above steps, the word order vectors for the C and the S of the above example C and S is listed below:

$r_1 = [1, 2, 3, 4]$

$r_2 = [4, 2, 3, 1]$

Thus, a word order vector is the basic structural information carried by the SPOs. We have used the formula as given in equation 12 for calculating the similarity of a C and an S, where r_1 and r2 represent the word order vector as we have just discussed.

$$S_r = 1 - \frac{\| \mathbf{r}_1 - \mathbf{r}_2 \|}{\| \mathbf{r}_1 + \mathbf{r}_2 \|}.$$

Equation 12 Word order semantic similarity calculation

That is, word order similarity is determined by the normalized difference of word order vectors. The final result which we got from our program for the S_r above is: 0.8235294.

In the following we show an example in which the two phrases don't contain exactly the same words, but do have the same (e.g., the Hit) and similar (e.g., Hurricane <-> Strom) word pairs:

C = Hurricane-> Hit-> heavy Loss

S = Storm-> Hit-> Building

The joint word set for these SPOs is

CS = {Hurricane, Hit, heavy, Loss, storm, building}

Similarly, the vectors of word order derive from the joint word set CS. So, the word order vectors calculated according to the above rules for C and S are listed below:

r_1 *= [1, 2, 3, 4, 1, 2]*

r_2 *= [1, 2, 1, 1, 1, 3]*

Based on equation 12, the word order semantic similarity between these two SPOs is

$$wordOrderSemValue\ = 0.84444445$$

4.3.2 Word order similarity between concept and SPO: An alternative method

In the future, we could also implement an alternate method for calculating the word order similarity between the concept and an SPO. In the new method, we concentrate to find the position of the word located at subject, predicate and object, rather than finding the position of all the words in an SPO, individually. We assign the numbers one, two, and three for the words which are present at subject, predicate, and object location respectively in a given concept.

For Example: suppose we have the concept, C and an SPO, S

C = boy -> loves -> his dog

S = Dog -> loves -> his boy

First, we derive the word order similarity vector r1 from C. For deriving the r1, we consider the word “boy,” which is present at the subject position. So, we assign it the number one. Then, we select the word present at predicate location “loves,” and assign it with number 2. Finally, we consider the words “his dog” present at object location, and assign it with number 3. The word order similarity vector r1 derive as below.

r1 = [1, 2, 3]

Now, we derive the word order similarity vector r2 from S. In this case, first, we consider the word “Dog” in S, which is present at subject position. Then, we find the word in C, which is more semantically similar for the consider word “Dog” in S. In this case, the “Dog” present in S at subject position is similar to the “dog” present in C at object position, so we assign the number 3 for “Dog” present in S. The word “loves” present in S at predicate position is the same as “loves” present in C. So, we assign the number 2 for the word “loves” in S. At last, the words “his boy” are present at object location in S, and the word “boy” is the same as the word “boy” in C, which is present at subject location. So, we assign unique number 1 for the word “boy” in S and ignore the word his. The word order similarity vector for r2 derives as below.

r2 = [3, 2, 1]

We use the same formula as given in equation 12 for calculating the similarity of the concept and an SPO, where r1 and r2 represent the word order similarity vectors, as we have just discussed.

$$S_r = 1 - \frac{\| \mathbf{r}_1 - \mathbf{r}_2 \|}{\| \mathbf{r}_1 + \mathbf{r}_2 \|}.$$

Thus, the final result is 0.83333

We consider the different example for the concept and an SPO in which the two SPOs don't contain exactly the same words, but do have the same word (e.g., the Hit) and semantically similar words (e.g., Hurricane <-> Strom):

C = Hurricane-> Hit-> heavy Loss

S = Storm-> Hit-> Building

Based on the above discussion, we create the word order similarity vector r1 for C as below

r1 = [1, 2, 3]

r1 = [1, 2, 3] (since building is more semantically similar with heavy)

Thus, the final result is 1.

4.4 Semantic similarity between words

This is the most important step for calculating the semantic similarity between a concept and an SPO. There are a number of methods available for measuring semantic similarity between words. As we have discussed above, we chose a Hybrid approach in this thesis work. Three methods are selected: two from knowledge based measures and the other from the PMI method for corpus based measures. As discussed above, among the knowledge based measure, the Lin [17] and Wu&Palmer [4] are two methods which are suitable for our work. To make our model be applicable to any domain in general, we included a corpus measure in our work to cover domain specific words. To apply the corpus based measure, a local dictionary is created in the domain of interest. This local dictionary serves to increase the efficiency of the program. The dictionary could be generated during the program's running time, but it would take more execution time. In our work, the local dictionary is generated off-line before the program's execution. We keep only those words which are not available in WordNet or Wikipedia in the local dictionary. In this way the number of words contained in our local dictionary is decreased from 6,565 to 1,359, where 6,565 is the number of total words extracted from the domain specific text corpus.

4.5 Semantic similarity between concept and SPO

The concept and the SPO in a given list are a collection of words, so we used the words of SPO to represent the concept or SPO. Using the method discussed above, first we calculated the lexical semantic vectors for both the concept and the SPO, denoted as S_1 and S_2 respectively

For calculating the lexical vector first created a matrix, whose columns were the joint set of words, and rows were words present in a concept or a SPO. So, for example, in calculating the S_1 lexical semantic vector, the words from the concept formed the rows and the joint set of words from both the concept and the SPO formed the columns of the matrix. Similarly, for calculating the S_2 lexical vector, the rows and the columns of the matrix came from the SPO and joint set, respectively. For example, if we have

Concept: *Cyberattack -> Cause -> economic Loss*

SPO: *Attacker -> Download -> information*

Then we have the joint set of words as

SP = {Cyberattack, cause, economic, loss, attacker, download, information}

Now, for calculating the lexical semantic vector, we created two matrixes, one for each vector as mentioned above.

The value in each cell entry of the matrix is determined by the semantic similarity between the words. For example, if we consider a joint set of words and a concept, then value of each cell of the matrix is obtained based on the following rules:

1. If the column words present in the concept, then the value of the cell is set to 1.
2. If the column word is not contained in the concept, a semantic similarity score is computed between the column and the row words based on the method given below.

For measuring the value between words, we have used both knowledge based methods and corpus method as discuss above. First, we calculated the semantic similarity value by using WorldNet's two methods and then took the average of these two methods.

$$finalAvgSemValueW = \frac{(semanticSimMeasureLIN\ +\ semanticSimMeasureWP)}{2}$$

Equation 13 Average value calculation from Lin and WP methods

Then, we computed the value of the semantic similarity between the words by using Wikipedia. Once we calculated the value from both the resources, then we calculated the final semantic similarity value based on these two values by using equation 14.

$$SemValue = (\,\sigma * \text{ finalAvgSemValueWN }) + \ (1 - \ \sigma) * \text{ finalAvgSemValueDisco }))$$

Equation 14 semantic similarity value from WordNet and Wikipedia resources

The value of σ is decided based on the feedback from the human evaluation. In this thesis work, the value of σ is set at 0.5 (i.e. the average of WordNet and Wikipedia scores)

If there are no matching words present in WordNet or Wikipedia, then the function will return to zero as the semantic similarity value. In this case, since the word is not present in both resources, we looked into the local dictionary. The semantic similarity measure from the local corpus is discussed below.

We set a threshold value as 0.2 for reducing noise. If the semantic similarity value drops below this threshold, then insert zero value in the matrix cell. The value of each dimension of the vector is the maximum value among all the values in that particular column. The overall similarity S_s is then computed based on the cosine similarity formula.

$$S_s = \mathbf{cos}(\boldsymbol{S_1},\ \boldsymbol{S_2}) = \frac{(\boldsymbol{S_1} \times \boldsymbol{S_2})}{|\boldsymbol{S_1}| \times |\boldsymbol{S_2}|}$$

Below is the example of calculating the lexical semantic vector.

	Cyberattack	Cause	Economic	Loss	Attacker	Download	Information
Cyberattack	1	0.367	0.0	0.428	0.0	0.0	**0.326**
Cause	0.367	1	0.0	0.544	0.294	0.0	**0.350**
Economic	0.0	0.0	1	0.223	0.0	0.0	**0.0**
Loss	0.428	0.544	0.223	1	0.209	0.0	**0.349**
$\breve{S}_1$	**1**	**1**	**1**	**1**	**0.294**	**0.0**	**0.349**

Table 3 Matrix for S_1 lexical vector

So, $\widetilde{\boldsymbol{S_1}}$ lexical vector is

$$\widetilde{\boldsymbol{S_1}} = \{1, 1, 1, 1, 0.294, 0.0, 0.349\}$$

Same as for lexical semantic vector $\widetilde{\boldsymbol{S_2}}$

	Cyberattack	**Cause**	**Economic**	**Loss**	**Attacker**	**Download**	**Information**
Attacker	0.0	0.294	0.0	0.209	1.0	0.0	**0.0**
Download	0.0	0.0	0.0	0.0	0.0	1.0	**0.0**
Information	0.326	0.35	0.0	0.349	0.0	0.0	**1.0**
S_2	**0.326**	**0.35**	**0.0**	**0.349**	**1.0**	**1.0**	**1.0**

Table 4 Matrix for S_2 lexical vector

So, $\widetilde{\boldsymbol{S_2}}$ lexical vector is

$$\widetilde{\boldsymbol{S_2}} = \{0.326, 0.35, 0.0, 0.349, 1, 1, 1\}$$

For calculating the semantic similarity, we used the cosine similarity function

$$S_s = \cos(s_1, s_2) = \frac{s_1 \cdot s_2}{|s_1| \times |s_2|}$$

Thus, the final semantic similarity value between the concept and the SPO is

$$\breve{S}_s = 0.4988554$$

Based on the same concept and SPO, we now compute the word order as:

$$wordOrderSemValue = 0.84671533$$

The combined similarity score represents the overall SPO similarity. Thus, the overall semantic similarity computed by the sum of semantic similarity and word order similarity is as given in equation 15.

$$finalSemValue = (\rho * SimValue) + (1 - \rho) * wordOrderSemValue))$$

Equation 15 final semantic similarity calculation from combination of semantic similarity and word order similarity

Where the SemValue is calculated by

$$SemValue = (\sigma * \text{finalAvgSemValueWN}) + (1 - \sigma) * \text{finalAvgSemValueDisco}))$$

For the example above, we have final semantic similarity value between the concept and the SPO is:

$$finalSemValue = (\,0.5\, *\, 0.499885\,) + \,(1 - \,0.5) *\, 0.84671533\,))$$

That is,

$$finalSemValue = 0.67330$$

Where $\rho\, \leq 1$ decides the relative contributions of semantic and word order information to the overall similarity computation. The value of ρ in our case is 0.5. We fixed this value based on human evaluation result for one dataset, and then for the rest of the dataset, we compared the score.

5 Evaluation and Experimental Results

Even though a few related papers have been published, there are currently no appropriate data sets available in our cyber security application domain for the evaluation of proposed method for comparing the given concept with an SPO list. Therefore, we built our own datasets for this thesis work. For building such a data set, we used news articles from general topics and the cyber security domain so that we could test our method in the general as well as in the specific domain. For constructing such a dataset, the following steps are taken:

(1) A set of news articles in the cyber security domain were selected.

(2) The selected news articles were converted to xml files,

(3) A SPO list was generated from the xml files by using a NLP tool [1].

(4) A concept SPO was identified from the list.

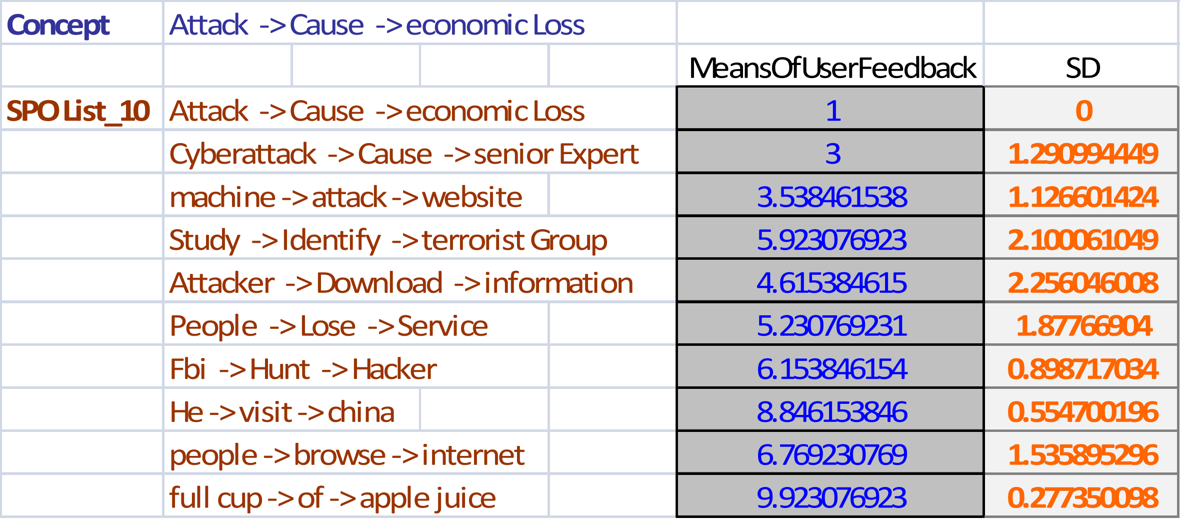
(5) A subset of the SPO list was selected again for comparing it with concept.

We prepared two datasets for comparison. Out of the two datasets, the first dataset contains 10 SPOs in a list and the other one contains 5 SPOs in list. The SPOs in each dataset were selected from the cyber security domain and the general domain. Some of the SPOs we selected were close to the given concept, and others were not close to the given concept.

5.1 Experimental Results Analysis

In order to evaluate our proposed method, we conducted several surveys based on the prepared datasets. We collected human feedback for the two datasets. The participants consisted of 39 volunteers for one dataset and 46 volunteers for another dataset including professors, students and common people who are not related to the specific field. In order to evaluate the method fairly, we selected people at different knowledge levels, including those who are more knowledgeable in the field and those who are less knowledgeable in the field. The participants were asked to compare the given concept with the SPOs lists and to rank them based on their relevancy. The SPOs were to be ranked in order of relevance by assigning a number from zero to one, with one being the most relevant to the given concept. We provided one example in the survey sheet to give participants an idea of how to rank the SPO list for a given concept. The order of appearance of SPOs in a dataset was randomized in each dataset. This was to avoid any biased decision being introduced by order of appearance of the SPOs in the list.

Table 5 is an example of the first dataset which has ten SPOs in a list to be compared for a given concept. We received response from a total 39 participants for this dataset. Then, we calculated the average and standard deviation of all the response received from the participants. Human similarity scores are provided as the mean score for each pair and have been scaled into the range in terms of the number of SPOs in the list. In this case, we scaled it from 1 to 10 ranges. In the raw data we collected from participants, we found for some of the SPO's standard deviation was higher which was likely caused by certain inputs that can be considered as outliers. These outliers were removed from the survey results to maintain a proper level of the standard deviation of the results.



Concept	Attack -> Cause -> economic Loss		
		MeansOfUserFeedback	SD
SPO List_10	Attack -> Cause -> economic Loss	1	0
	Cyberattack -> Cause -> senior Expert	3	1.290994449
	machine -> attack -> website	3.538461538	1.126601424
	Study -> Identify -> terrorist Group	5.923076923	2.100061049
	Attacker -> Download -> information	4.615384615	2.256046008
	People -> Lose -> Service	5.230769231	1.87766904
	Fbi -> Hunt -> Hacker	6.153846154	0.898717034
	He -> visit -> china	8.846153846	0.554700196
	people -> browse -> internet	6.769230769	1.535895296
	full cup -> of -> apple juice	9.923076923	0.277350098

Table 5 First Dataset

Due to a lack of reference sources for comparing the accuracy of the output of the proposed method, we decided to performed statistical analysis. In this analysis, we measured T-stat value for T-test, P-value, and Pearson Correlation. We also considered null hypothesis for comparing the result of the proposed method and the human evaluation mean. In the following section, we discussed briefly about the Hypothesis test, T-test and P-value.

5.2 Statistical Analysis

As we know, statistical analysis is very useful when we look for differences that are small compared to the imprecision and the human cognition variability. In statistical analysis, generally we want to conclude from the set of data which we collect from human surveys, to make general conclusions. In order to properly anticipate a statistical analysis, we considered the hypothesis test. It is a very useful tool to evaluate the sample of data collected from subjects and helps in order to make decisions based on the data.

There are two types of hypothesis tests which can be perform. The first one is the null hypothesis test and the second is the alternative hypothesis Test. The null hypothesis test states that no difference exists between two samples of data; the results obtained from different ways are same. The null hypothesis statement is presumed to be true until statistical evidence nullifies it or reject it for an alternative hypothesis. An alternative hypothesis is a hypothesis which states that there is a difference between the two samples of data. In the hypothesis test, first we need to state the hypothesis statement. Second, we formulate the analysis plan. The third step is to analyze the data, and then, in the final step, we interpret the results. For analyzing the data set, the researcher generally calculates the p-value and the T-stat value.

p-value [23] measure the significance of a hypothesis test. It is the level of marginal significance within a statistical hypothesis test, representing the probability of the occurrence of a given event. The p-value is used as an alternative to rejection points to provide the smallest level of significance at which the null hypothesis would be rejected. The smaller the p-value, the stronger the evidence is in favor of the alternative hypothesis. P-values are generally considered significant if they are less than 0.05.

The T-test [22] is another method to perform a statistical examination of two dataset means. A T-test is used to determine whether a set or sets of scores are from the same population. There are three types of the T-test:

- **One-sample t-test:** Used to compare a sample mean with a known population mean or some other meaningful, fixed value.
- **Independent samples t-test:** Used to compare two means from independent groups.
- **Paired samples t-test:** Used to compare two means that are repeated measures for the same participants – scores might be repeated across different measures or across time.

In our case, first we considered that there is no difference between the proposed method output and the human evaluation means. For the second step, we considered that the significance level equal to 0.05 means that if P- value is greater than 0.05, then we will fail to reject the null hypothesis. We calculated the means of human evaluation which we got from 39 participants' responses. Since the variance in the population were unknown means that the entire pool from which we drawn the statistical sample or data were unpredictable and our sample size was also small, so we used the T-test. Since the two variables, which we compared, were related to the same SPO for each row, so we used paired samples T-test (a dependent T-test). We also calculated the p – value based on our sample data.

Testing data were paired because they are performed on the same samples or subjects. Since we compared the two sets of data, one is output of the proposed method and another is the mean of user responses for the same SPO, the expectation was that the two values should not reject the null value hypothesis. The acceptance of the null hypothesis (or failing to reject it) implies that there is no significant difference between the scores obtained from our integrated method and the means of human evaluation. This result

suggests that the ranking provided by the integrated method cannot be identified apart from a corresponding human ranking of the SPO triples. We used Microsoft Excel for performing the paired T–test, and drawing the chart between the two scores.

We also considered Pearson Correlation as another factor in order to compare our results more closely with the mean of human evaluation. The correlation between two variables reflects the degree to which the variables are related. It is widely used in the sciences as a measure of the strength of linear dependence between two values. The absolute value of both the sample and population Pearson correlation coefficients are less than or equal to 1. Correlations equal to 1 or -1 correspond to data points lying exactly on a line. So, the higher the value of Pearson Correlation reflects more strengths of the linear association.

5.3 Evaluation of the results

In the following section, we discuss how outputs of the integrated measuring results vary by changing the values of the co-efficient ($\rho \; and \; \sigma$). We tested our model's output for the six different conditions. These conditions are (1) measures with or without including the WordNet, (2) measures with or without including the Wikipedia, and (3) measures with or without including the Word Order similarity method. The values of constant coefficients ρ and σ are adjusted according to the results of the comparison between the proposed method with the human survey outcomes. Comparisons between the proposed method output and the human evaluation results for each condition are shown in figures 11 to 15. The charts in these figures give ideas of how program's output varies in different conditions. After carefully analyzing all the results, we finally selected the optimal values of these co-efficient.

First Test: We considered the contribution of the word order similarity is 0.47 and semantic information is 0.53. For our other factors, we considered only Wikipedia and the local corpus as resources, and we didn't consider the WordNet. Then we compared the score of the proposed method with the means of human evaluation and drew the chart in excel to compare how much accuracy we achieved without considering WordNet similarity scores. Figure 11 shows the comparison chart between proposed method output and the mean of the human evaluation.

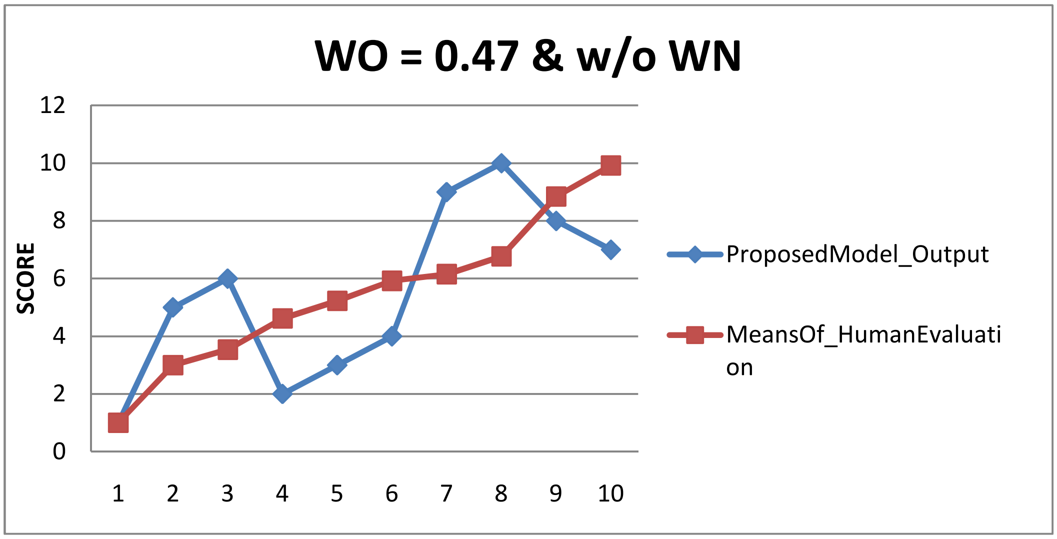


Figure 11 Results comparison between proposed method output and Human Evaluation mean

We performed the T-test for the statistical analysis and calculated the T-stat and the p-value in order to infer our conclusion regarding the null hypothesis. Table 6 shows the results. As we can see that the t- critical one tail is greater than the T-stat, so we failed to reject the null hypothesis. It means that there is no significant difference between the scores of the two methods. Also, we found that the p-value is also greater than the alpha value, which we considered as 0.05 as a significance level, so again we failed to reject the null hypothesis. This result suggests that the ranking provided by the integrated method cannot be identified apart from a corresponding human ranking of the SPO triples.

Based on Pearson correlation factor, we tried to select the values of the co-efficient. In this case, we found person correlation is 0.641149, which is low compared to the other tests' results as mentioned in rest of this section.

t-Test: Paired Two Sample for Means		
	ProposedModel_Output	*MeansOf_HumanEvaluation*
Mean	5.5	5.5
Variance	9.166667	7.131164
Observations	10	10
Pearson Correlation	0.641149	
Hypothesized Mean Difference	0	
Df	9	
t Stat	1.3E-10	
P(T<=t) one-tail	0.5	
t Critical one-tail	1.833113	
P(T<=t) two-tail	1	
t Critical two-tail	2.262157	

Table 6 T-test result for chart WO = 0.47 & w/o WN

Second Test: In this case, we considered the resources including the local corpus and the WordNet but not including the Wikipedia. Comparison chart is shown in figure 12. Table 7 represents the statistical calculation values for T-test and p-value.

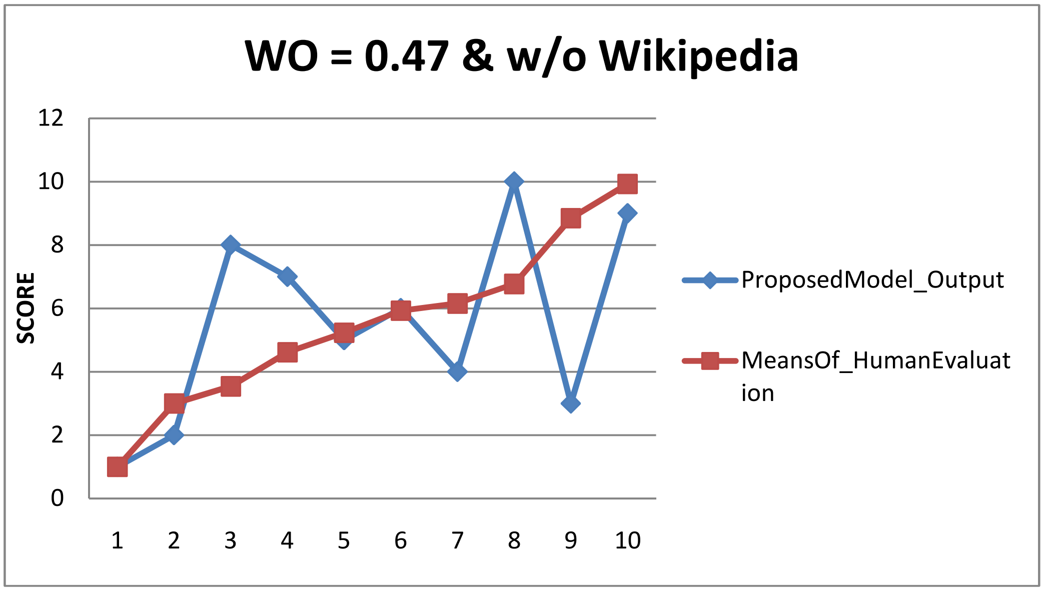


Figure 12 Results comparison between proposed method output and Human Evaluation mean

t-Test: Paired Two Sample for Means		
	ProposedModel_Output	*MeansOf_HumanEvaluation*
Mean	5.5	5.5
Variance	9.166667	7.131164

Observations	10	10
Pearson Correlation	0.480465	
Hypothesized Mean Difference	0	
df	9	
t Stat	1.08E-10	
P(T<=t) one-tail	0.5	
t Critical one-tail	1.833113	
P(T<=t) two-tail	1	
t Critical two-tail	2.262157	

Table 7 T-test result for chart WO = 0.47 & w/o Wikipedia

Again in this case, T-stat value is less than the t-critical value and the p-value is 0.5 for one tail and 1 for two tails, which is greater than the alpha value (0.05), so we failed to reject the null hypothesis, this result again suggests that the ranking provided by the integrated method cannot be identified apart from a corresponding human ranking of the SPO triples. Even though results we achieved in this case are good, but the Pearson Correlation value is lower than the above test value which is 0.480465, so we concluded that not considering Wikipedia as one of the resources leads to poor scores of the proposed method.

Third Test: we conducted the following test with only the Word order similarity measure. In this case, we didn't consider the semantic similarity measure. The compared chart is shown in figure 13 and the statistical calculation values are shown in table 8.

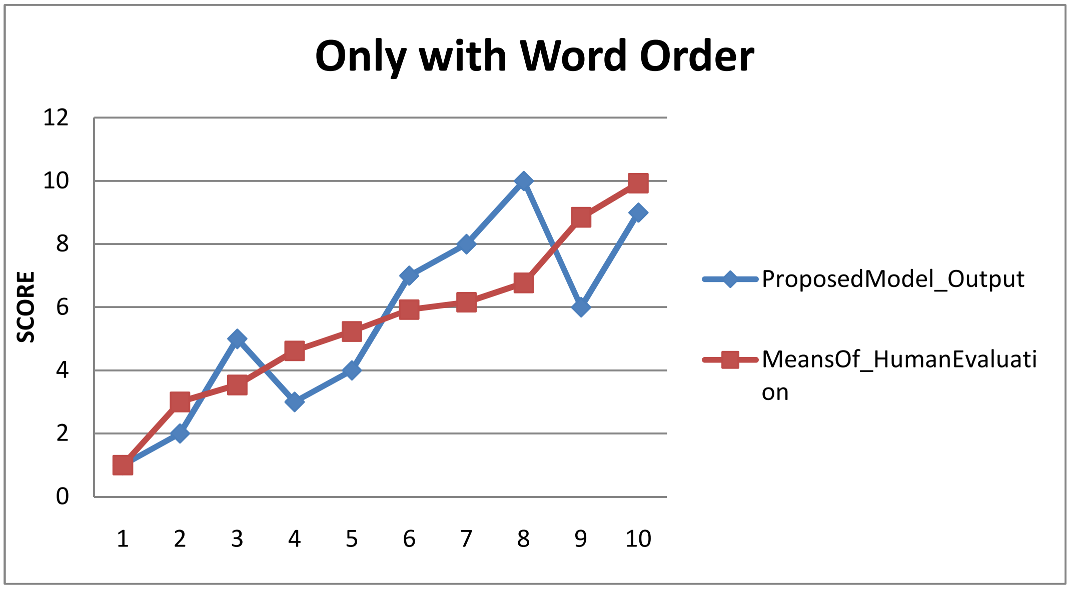


Figure 13 Results comparison between proposed method output and Human Evaluation mean

t-Test: Paired Two Sample for Means		
	ProposedModel_Output	*MeansOf_HumanEvaluation*
Mean	5.5	5.5
Variance	9.166667	7.131164
Observations	10	10
Pearson Correlation	0.793376	
Hypothesized Mean Difference	0	
df	9	

t Stat	1.7E-10
P(T<=t) one-tail	0.5
t Critical one-tail	1.833113
P(T<=t) two-tail	1
t Critical two-tail	2.262157

Table 8 T-test result for chart Only with Word Order

In this case also, the T-stat value is less than the T-critical value and the p-value is greater than the alpha value, so we failed to reject the null hypothesis. This means that both the results we achieved from the proposed method and the human evaluation are good and reliable. In this case, the Pearson Correlation value we got is 0.793376, which is higher than the above tests shown in table 7, but lower than the absolute value one, so we concluded that only the word order similarity measure is not sufficient for comparing the concept with a SPOs list.

Fourth Test: In the fourth test, we didn't consider the word order similarity measure, and we only considered the semantic similarity measure. Figure 14 shows the comparison chart and table 9 shows the statistical calculations.

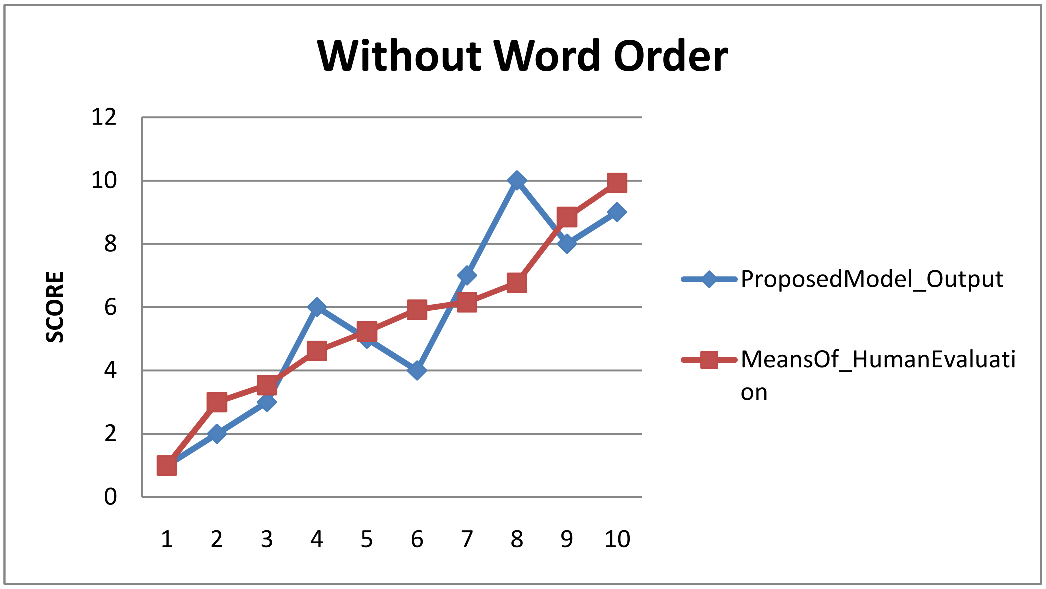


Figure 14 Results comparison between proposed method output and Human Evaluation mean

t-Test: Paired Two Sample for Means		
	ProposedModel_Output	*MeansOf_HumanEvaluation*
Mean	5.5	5.5
Variance	9.166667	7.131164
Observations	10	10
Pearson Correlation	0.87266	
Hypothesized Mean Difference	0	
df	9	
t Stat	2.14E-10	
P(T<=t) one-tail	0.5	

t Critical one-tail	1.833113
P(T<=t) two-tail	1
t Critical two-tail	2.262157

Table 9 T-Test result for chart Without Word Order

Even though we didn’t consider the word order measure, still we got t- stat value less than t- critical values and p-value greater than alpha value, so we failed to reject the null hypothesis. This result again suggests that the ranking provided by the integrated method cannot be identified apart from a corresponding human ranking of the SPO triples. In this case, the Pearson Correlations value is higher than the other tests but still less than 1, so we conclude that including the word order similarity measure can improve the results

Fifth Test: The following test was performed by considering all the factors while giving the following weight to each factors; WordNet 0.47, Wikipedia 0.53, and word order 0.45. The comparison results show in figure 15 and table 10 shows the statistical calculations.

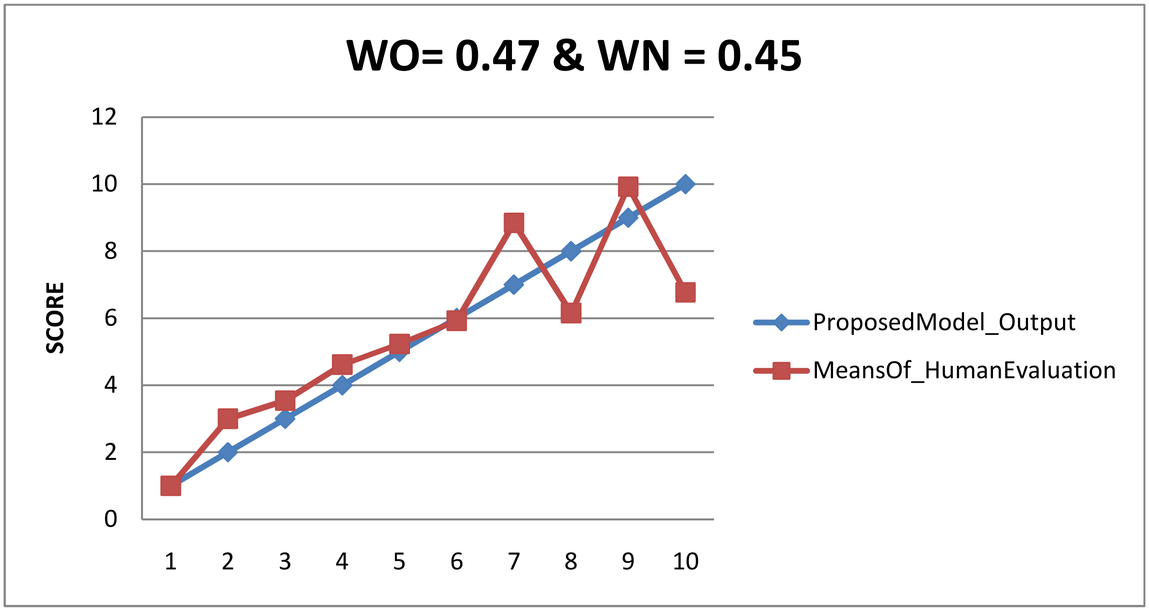


Figure 15 Results comparison between proposed method output and Human Evaluation mean

t-Test: Paired Two Sample for Means		
	ProposedModel_Output	*MeansOf_HumanEvaluation*
Mean	6	6
Variance	7.5	5.210059
Observations	9	9
Pearson Correlation	0.818326	
Hypothesized Mean Difference	0	
Df	8	
t Stat	-1.9E-16	
P(T<=t) one-tail	0.5	

t Critical one-tail	1.859548
P(T<=t) two-tail	1
t Critical two-tail	2.306004

Table 10 T-test result for chart WO= 0.47 & WN = 0.45

In this case, we again found that we failed to reject the null value hypothesis based on the statistical T-test values, and the Pearson Correlation value is much less than one. This result again suggests that the ranking provided by the integrated method cannot be identified apart from a corresponding human ranking of the SPO triples.

Sixth Test: In this test, we gave equal weight to all the parameters. For example, we considered the value of both the coefficients (ρ, σ) are 0.5 and ran the test. Figure 16 shows the caparison chart between the proposed method results and the human evaluation mean. Table 11 shows the statistical calculations.

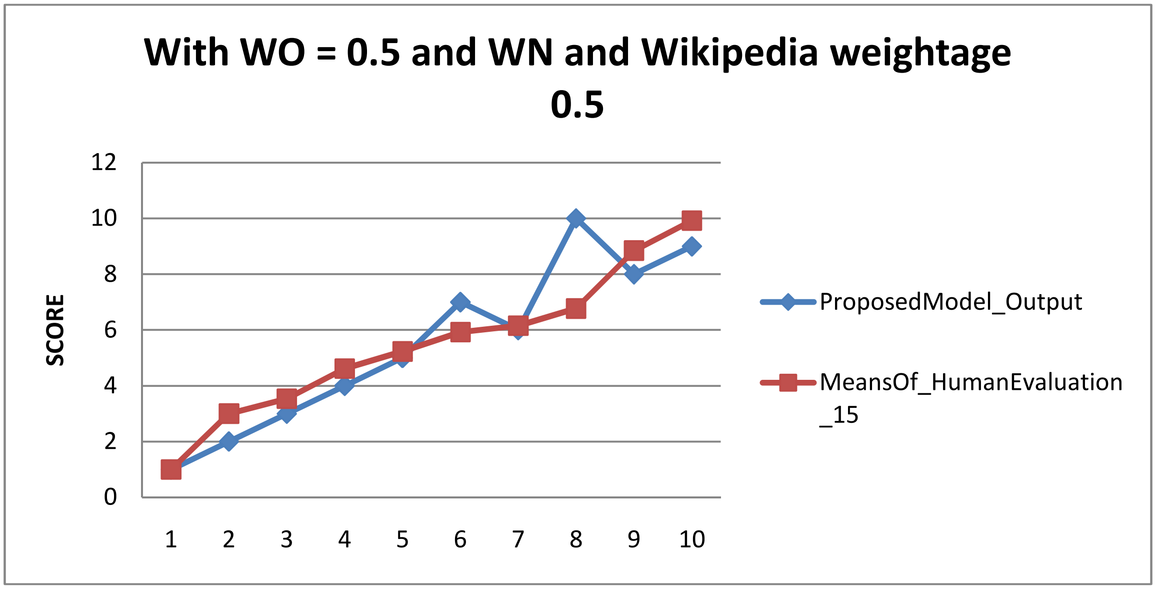


Figure 16 Final results comparison between proposed method output and Human Evaluation mean

t-Test: Paired Two Sample for Means		
	ProposedModel_Output	*MeansOf_HumanEvaluation*
Mean	5.5	5.5
Variance	9.166667	7.131164
Observations	10	10
Pearson Correlation	0.905431	
Hypothesized Mean Difference	0	
df	9	
t Stat	2.46E-10	
P(T<=t) one-tail	0.5	

t Critical one-tail	1.833113
P(T<=t) two-tail	1
t Critical two-tail	2.262157

Table 11 T-test result for chart where WO has a weight of 0.5 and WN and Wikipedia each have weight of 0.5

This is the best result we obtained. In this case, again we failed to reject the null hypothesis based on the statistical calculations; therefore we can say that there is no significance difference between these two methods. This result again suggests that the ranking provided by the integrated method cannot be identified apart from a corresponding human ranking of the SPO triples.. Secondly, we obtained the highest Pearson Correlations value in this case, 0.905431. Based on all the analyses; we decided to set the co-efficient values of 0.5.

Using these co-efficient values, we compared our results in the same dataset with the responses of 39 participants. We achieved the Pearson Correlation value 0.910269, which is greater than the previous values.

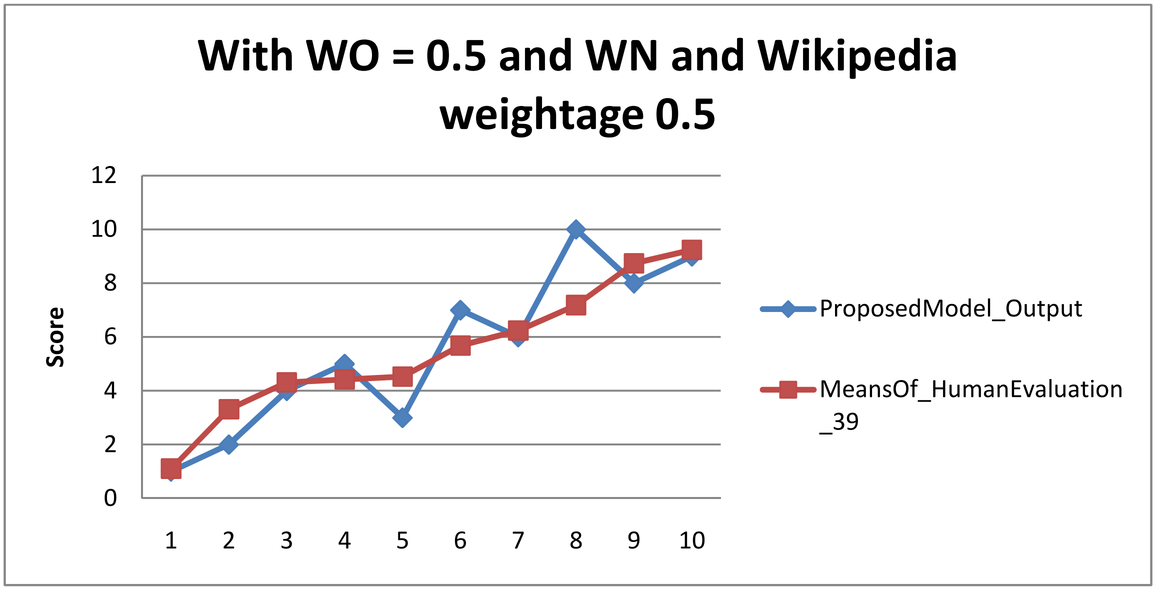


Figure 17 Results comparison between proposed method output and Human Evaluation mean 7

t-Test: Paired Two Sample for Means_39		
	ProposedModel_Output	*MeansOf_HumanEvaluation_39*
Mean	5.5	5.476316
Variance	9.166667	6.160888
Observations	10	10
Pearson Correlation	0.910269	
Hypothesized Mean Difference	0	
Df	9	
t Stat	0.058373	
P(T<=t) one-tail	0.477364	

t Critical one-tail	1.833113
P(T<=t) two-tail	0.954727
t Critical two-tail	2.262157

Table 12 T-Test result for chart With WO = 0.5 and WN and Wikipedia weightage 0.5 each

Figure 18 shows the second dataset with 5 SPOs in the list. For this dataset, we collected 46 responses in our survey from participants that included student and professors and non profession people in this domain.

Concept recent Attack -> On -> Internet

SPO List_5		MeansOf_HumanEvaluation_46	SDV
	Cyberattack -> Cause -> security problem	**1.239130435**	**0.705054217**
	Attack -> Alarm -> government Official	**2.173913043**	**0.797338568**
	Individual -> Expect -> Communication	**3.347826087**	**0.87476705**
	Topic -> Raise -> strong Passion	**3.608695652**	**0.649042401**
	quick brown dog -> jumps -> over foxg	**4.630434783**	**0.903295095**

Figure 18 second dataset with 5 SPOs in the list

In this test also, we gave equal weight to all the parameters. For example, we considered the value of both the coefficients (ρ, σ) are 0.5 and ran the test. Figure 19 shows the caparison chart between the proposed method results and the human evaluation mean. Table 13 shows the statistical calculations.

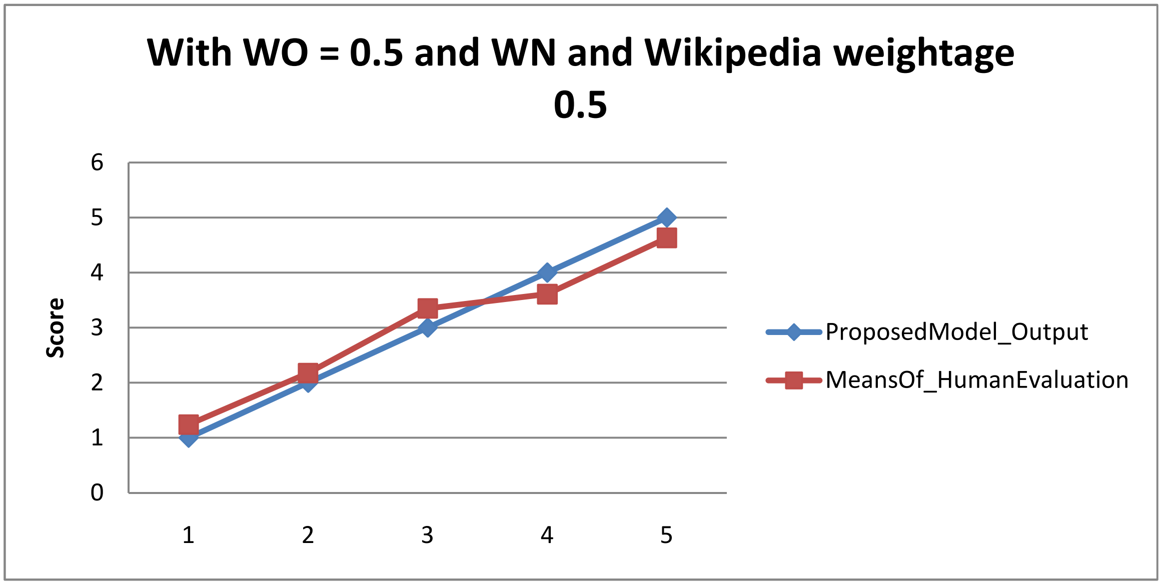


Figure 19 Second dataset result comparison with human evaluation

t-Test: Paired Two Sample for Means		
	ProposedModel_Output	*MeansOf_HumanEvaluation*

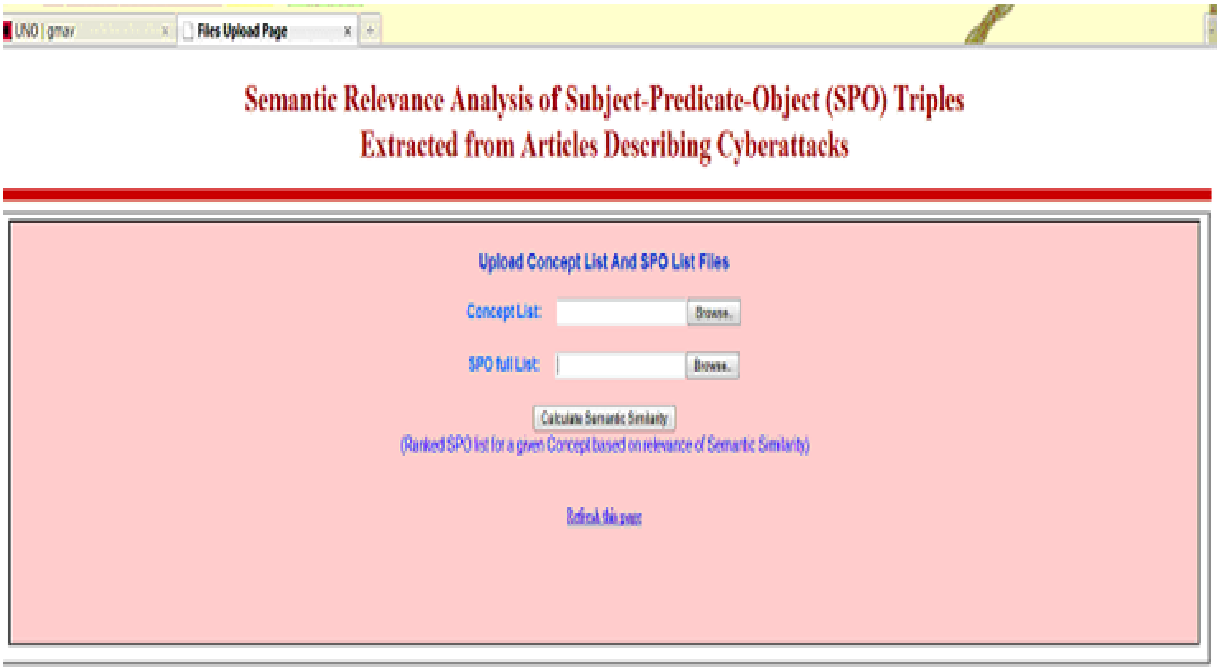
Mean	3	3
Variance	2.5	1.733223062
Observations	5	5
Pearson Correlation	0.986908167	
Hypothesized Mean Difference	0	
df	4	
t Stat	2.81399E-16	
P(T<=t) one-tail	0.5	
t Critical one-tail	2.131846782	
P(T<=t) two-tail	1	
t Critical two-tail	2.776445105	

Table 13 T-Test result for second dataset With WO = 0.5 and WN and Wikipedia weightage 0.5

In this case, again we failed to reject the null hypothesis based on the statistical calculations; therefore we can say that there is no significance difference between these two methods. This result again suggests that the ranking provided by the integrated method cannot be identified apart from a corresponding human ranking of the SPO triples. The Pearson Correlations value we got in this case is 0.986908167.

5.4 Implementation

We have also developed a web application user interface for the proposed method. Using this web application interface, user can upload the concept.txt file, which will contain one or more concepts and the SPOList.txt file, which will have a list of SPOs that need to be ranked with all the given concepts. Once the user uploads the files, the system will return with the ranked SPOs based on their relevance of semantic similarity. Figure 20 shows the screen shots of the web application. Figure 21 shows the result page in which the system display once the user uploads the concept and SPOs list files.

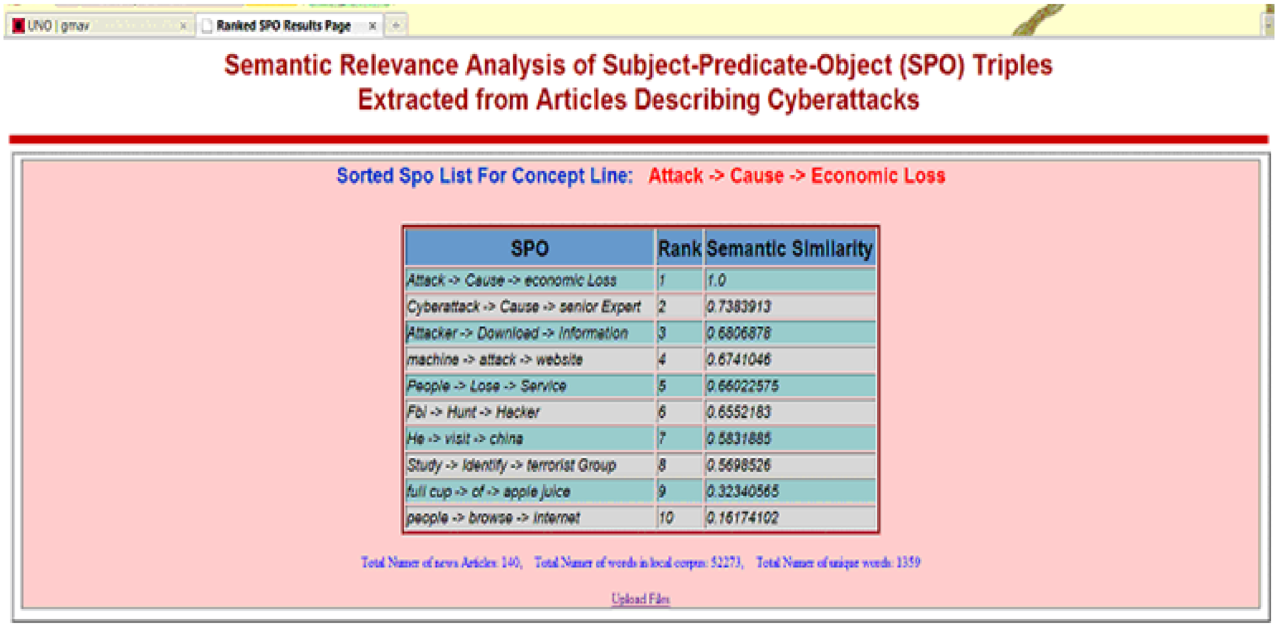


Semantic Relevance Analysis of Subject-Predicate-Object (SPO) Triples
Extracted from Articles Describing Cyberattacks

Upload Concept List And SPO List Files

(Ranked SPO list for a given Concept based on relevance of Semantic Similarity)

Figure 20 Web application user interface



Semantic Relevance Analysis of Subject-Predicate-Object (SPO) Triples Extracted from Articles Describing Cyberattacks

Sorted Spo List For Concept Line: Attack -> Cause -> Economic Loss

SPO	Rank	Semantic Similarity
Attack -> Cause -> economic Loss	1	1.0
Cyberattack -> Cause -> senior Expert	2	0.7383913
Attacker -> Download -> Information	3	0.6806878
machine -> attack -> website	4	0.6741046
People -> Lose -> Service	5	0.66022575
Fbi -> Hunt -> Hacker	6	0.6552183
He -> visit -> china	7	0.5831885
Study -> Identify -> terrorist Group	8	0.5698526
full cup -> of -> apple juice	9	0.32340565
people -> browse -> internet	10	0.16174102

Total Numer of news Articles: 140, Total Numer of words in local corpus: 52273, Total Numer of unique words: 1359

Upload Files

Figure 21 Ranked SPOs list for a given concept

For constructing the interface, we used Java, Java Servlet, JavaScript, and HTML in order to create the web application page. We used the Apache Tomcat 5.5.25 server for client server communication. NetBeans 7.0, we used as IDE for developing the whole java web application.

6 Conclusions, contribution and future work

This paper has presented an integrated approach for determining the similarity between a given concept and an SPO based on the semantic and syntactic information they contain. The proposed method is based on the work of [9] [10], with improvement by incorporating the local dictionary scheme. Using this method, we can measure the semantic similarity between a concept and a list of SPOs or a short text, or between two sentences based on their semantic and word order information. For calculating the semantic similarity among the words, we considered three resources. First is the lexical database WordNet in English, second is Wikipedia in English, and the third is the local corpus. The WordNet and Wikipedia cover all the set of general purpose words and the local corpus covers all the domain specific words that were either not contained in WordNet and Wikipedia, or having particular meaning or usage in the particular domain of concern. Thus, the local corpus reflects the actual usages of the language and the words for a specific domain. Thus, our semantic similarity not only captures common human knowledge, but is also able to adapt to an application area by using a corpus specific to that application. For the local corpus application, we used the SOC-PMI method, which determined the semantic similarity of two words, even though they do not have co-occurrence in the corpus. Actually, we considered a second order co-occurrence of the words in processing. The method takes judgment of co-occurrences not only of the words themselves, but also an extension to their neighboring words. That is, the co-occurrence of the words through their neighbors indirectly. We also considered the semantic impact of the word order on the SPO's meaning. The overall semantic similarity between a concept and each SPO in a list is then calculated by a combination of the primary semantic similarity and the word order syntactic similarity. For evaluating the proposed method, we prepared our own datasets and collected the feedback by a human survey. We completed our survey over 40 users, compared the human survey result with the model executions, and then analyzed both the results with the T-test and the p-value evaluation. Based on the T-test and P-value, we failed to reject the null hypothesis. So, we concluded that there is no significant difference between the proposed method and the human evaluation.

6.1 Main Contribution

The main contributions of this work are the following:

1. Ranked the SPO list for a given concept based on their relevance of semantic similarity.
2. For calculating the semantic similarity between a pair of words, we used three resources: WordNet, Wikipedia, and Local Corpus.
3. We combined formulas from available resources to get the optimum results. We selected Lin and Wu & Palmer from Knowledge Based Measure and PMI method for Corpus based Measure.
4. We created our own datasets for human survey in order to get feedback from the human and compare it for the proposed method results. From first dataset comparison, we determined the value of constant $\rho \ and \ \sigma.$ And for other dataset, we compared the output of proposed method's result and human feedback.
5. We applied statistic analysis to compare the proposed method's results with the human evaluation. We performed T-test and P- value in order to decide to consider or reject the Null Hypothesis.

6. We developed GUI for user to upload the concept and a SPO list files in order to get ranked SPOs list based on their relevance of semantic similarity for a given concept.

6.2 Future Work

For increasing the efficiency of the proposed method algorithm, we may create the PMI lookup table from the local corpus in advance, so that for calculating the semantic similarity between two words, we can avoid all the calculations at run time. Currently, the system takes 5 to 8 seconds for processing each pair of words from the local corpus. Though we improved the execution time with respect to previous execution time, which used to take 45 to 50 seconds for one paired of words. We used 32-bit operating system with Inter(R) Core™2 Duo CPU, 2.20 GHz frequency, and 2 GB of memory.

For improving the accuracy of the results, we may consider a further refinement of the S (subject) and O (Object) phrases such that the the adjectives in the phrases can be identified. By separating the head nouns and objects from the phrases and ignoring the adjectives in the similarity computation we could further reduce the noises caused by the adjectives and give more important sense recognition to those main noun words rather than to those supporting words.

We can also consider the word sense disambiguation to get more contextual information using the surrounding words.

We may intend to extend the analysis to find the reason of cause and effect from the document coming from natural language text. We may also use this work to match a set of documents related to given concept.

7 Reference

[1] William Sousan, Robin Gandhi, Qiuming Zhu, William Mahoney, "*Constructive Ontology Engineering*", ProQuest LLC, 2010.

[2] Wei Liu, Albert Weichselbraun, Arno Scharl, Elizabeth Chang, "*Semi-Automatic Ontology Extension Using Spreading Activation," Journal of Universal Knowledge Management*, Journal of Universal Knowledge Management 0(1), 2005, pp: 50-58.

[3] C. Leacock and M. Chodorow, "Combining local context and WorldNet similarity for word sense identification," *WordNet: An Electronic Lexical Database*, Volume 49(2), 1998, pp: 265–283.

[4] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Morristown, NJ, USA, 1994. pp: 133–138.

[5] Vincent D. Blondel et al., "A measure of similarity between graph vertices: Applications to synonym extraction and web searching" *SIAM Rev*., Volume 46, Issue 4, 2004, pp: 647 – 666.

[6] Dekang, Lin, "An information-theoretic definition of similarity," *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp: 296 – 304.

[7] Philip, Resnik, "Using information content to evaluate semantic similarity," *International Joint Conference on Artificial Intelligence* Montreal, Quebec, Canada, 1995, pp: 448-453.

[8] Owl web ontology language overview, w3c recommendation, 10 February 2004. See http://www.w3.org/TR/owl-features/, (as of July, 20, 2010).

[9] Aminul Islam and Diana Inkpen, " Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity," *ACM Transactions on Knowledge Discovery from Data (TKDD)* Volume 2, Issue 2, July 2008, Article 10.

[10] Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett, "Sentence Similarity Based on Semantic Nets and Corpus Statistics," *IEEE Transactions on Knowledge and Data Engineering* Volume 18, Issue 8 (August 2006), pp: 1138 – 1150.

[11] Thanh Ngoc Dao, Troy Simpson, "Measuring Similarity between sentences," *http://www.codeproject.com/KB/string/semanticsimilaritywordnet.aspx* , (as of July, 20, 2010).

[12] Yuhua Li, Zuhair A. Bandar, and David McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources," *IEEE Transactions on Knowledge and Data Engineering* Volume 15, Issue 4 (July 2003), pp: 871-882.

[13] http://wordnet.princeton.edu/ (As of January, 2011).

[14] Michael Mohler and Rada Mihalcea, “Text-to-text Semantic Similarity for Automatic Short Answer Grading,” *EACL '09 Proceedings of the 12th Conference of the European* 2009, pp: 567-575.

[15] http://www.linguatools.de/disco/disco.html (as of January, 2011).

[16] Sulema Torres and Alexander Gelbukh, “Comparing Similarity Measures for Original WSD Lesk Algorithm,” *A. Buchmann (Ed.) Advances in Computer Science and Application 43,* 2009, pp. 155-166.

[17] Philip Resnik, “Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language” *Journal of Artificial Intelligence Research*, Vol. 11, 1999, pp. 95-130.

[18] Budanitsky, Alexander and Hirst, Graeme, “Evaluating WordNet-based Measures of Lexical Semantic Relatedness,” *Computational Linguistics,* Volume 32, Issue 1, March 2006, pp: 13-47

[19] Aminul Islam, Diana Inkpen, “Second Order Co-occurrence PMI for Determining the Semantic Similarity of Words,” *Proceedings of the International Conference on Language Resources and Evaluation* (LREC 2006), Genoa, Italy, May 2006, pp. 1033-1038.

[20]http://www.codeproject.com/KB/string/semanticsimilaritywordnet.aspx?display=Mobile&fid=220882&df=90&mpp=25&noise=3&sort=Position&view=Quick&fr=51 (as of February, 2011).

[21] http://www.graphpad.com/articles/pvalue.htm (as of January 2011).

[22] http://en.wikiversity.org/wiki/T-test (as of January 2011).

[23] http://www.investopedia.com/t (as of January 2011).

[24] http://en.wikipedia.org/wiki/Tf%E2%80%93idf (as of March 2011).

[25] Palakorn Achananuparp, Xiaohua Hu and Xiajiong Shen, "The Evaluation of Sentence Similarity Measures," *10th international conference on Data Warehousing and Knowledge Discovery,* Volume 5182, Issue 2008, pp: 305-316.

[26] C.T. Meadow, B.R. Boyce, and D.H. Kraft, "Text Information Retrieval Systems," *second ed. Academic Press*, 2000.

[27] N. Okazaki, Y. Matsuo, N. Matsumura, and M. Ishizuka, "Sentence Extraction by Spreading Activation through Sentence Similarity," *IEICE Trans. Information and Systems*, vol. E86D, Issue 2003, pp: 1686-1694.