Sentence Similarity Measuring by Vector Space Model

U. L. D. N. Gunasinghe [1], W. A. M. De Silva[2], N. H. N. D. de Silva [3], A. S. Perera [4], W. A. D. Sashika[5], W. D. T. P. Premasiri[6]

Department of Computer Science and Engineering, University of Moratuwa
Moratuwa, Sri Lanka.

[1]nadeeshaan.10@cse.mrt.ac.lk
[2]anushka.10@cse.mrt.ac.lk
[3]nisansa@cse.mrt.ac.lk
[4]shehan@cse.mrt.ac.lk
[5]dulanga.10@cse.mrt.ac.lk
[6]thilina.10@cse.mrt.ac.lk

Abstract— **In Natural Language Processing and Text mining related works, one of the important aspects is measuring the sentence similarity. When measuring the similarity between sentences there are three major branches which can be followed. One procedure is measuring the similarity based on the semantic structure of sentences while the other procedures are based on syntactic similarity measure and hybrid measures. Syntactic similarity based methods take into account the co-occurring words in strings. Semantic similarity measures consider the semantic similarity between words based on a Semantic Net. In most of the time, easiest way to calculate the sentence similarity is using the syntactic measures, which do not consider grammatical structure of sentences. There are sentences which have the same meaning with different words. By considering both semantic and syntactic similarity we can improve the quality of the similarity measure rather than depending only on semantic or syntactic similarity. This paper follows the sentence similarity measure algorithm which is developed based on both syntactic and semantic similarity measures. This algorithm is based on measuring the sentence similarity by adhering to a vector space model generated for the word nodes in the sentences. In this implementation we consider two types of relationships. One of them is relationship between verbs in the sentence pairs while the other one is the relationship between nouns in the sentence pairs. One of the major advantages of this method is, it can be used for variable length sentences. In the experiment and results section we have been included our gain with this algorithm for a selected set of sentence pairs and have been compared with the actual human ratings for the similarity of the sentence pairs.**

Keywords— **Sentence Similarity, StanfordCoreNLP, Word Similarity, Semantic Similarity, Syntactic Similarity, WordNet**

I. Introduction

Today in most of the computer applications, text mining and Natural Language Processing are widely used. In those applications, underlying behaviour is based on the basic sentence similarity measuring components. In sentence similarity, measuring the fundamental idea is based on measuring the word similarity measures. The accuracy of the sentence similarity measures mostly rely on how the words are chosen in order to compare and the types of grammatical relations used in the process.

There are three major types of sentence similarity measuring techniques. Those are semantic similarity measures, Syntactic similarity measures and the hybrid similarity measures. In semantic similarity measures, those only consider about the semantic relationships of the sentences while the syntactic measures consider about the syntactic relationships. In hybrid methods, we consider both syntactic and semantic similarity measures and the output is a weighted combination of semantic and syntactic measures.

In this paper, we have described about the implementation of a sentence similarity-measuring algorithm proposed at [1]. During the implementation of the algorithm, there was a need of identifying a suitable lexical database and a text parser. For this purposes we used Wordnet 2.1 [4] Lexical Database and StanfordCoreNLP [2] text parser. Instead of using StanfordParser we shifted to StanfordCoreNLP Parser due to the enhanced capabilities of lemmatizing features. For the preprocessing purposes, we followed stemming with the help of SnowBall [5] Stemmer library and the StanfordCoreNLP's lemmatizing features.

II. Dependency Grammar and Part of Speech (POS) Tagging

In dependency grammar, which was proposed by Lucien Tesnière [3] mainly focuses about how the words are related to each other with identified grammatical relationships based on the verbs as the structural centre. In order to identify the correct set of Dependencies between the syntactical units of a sentence, it is necessary to identify the correct role of each syntactic unit in a given sentence. This is

978-1-4799-7732-1/14/$31.00 ©2014 IEEE

because, for a given set of verbs, nouns, adverbs, adjectives, etc. (Syntactical units) the role they play in a sentence can be changed according to the way those units are ordered. Therefore, it is necessary to identify the dependencies between the syntactical units, before following Natural Language Processing tasks.

In this implementation of the algorithm proposed in [1] we need to accurately identify the proper Part of Speech of each and every word in a sentence and then identify the grammatical dependencies of the words according to the arrangement of these syntactic units.

Identifying the correct dependencies and the POS tags in a given sentences we have used Stanford Parser and the implementation follows the Stanford Dependency Structure. In Stanford Dependency Structure, there are approximately 50 grammatical relationships. In the implementation process, we have used Penn Tree Bank part-of-speech tags and phrasal labels. Bellow sentence describe how Stanford Parser represent Dependency relationships.

“Bell, based in Los Angeles, makes and distributes electronic, computer and building products”.

- nsubj(makes-8, Bell-1)
- nsubj(distributes-10, Bell-1)
- vmod(Bell-1, based-3)
- nn(Angeles-6, Los-5)
- prep in(based-3, Angeles-6)
- root(ROOT-0, makes-8)
- conj and(makes-8, distributes-10)
- amod(products-16, electronic-11)
- conj and(electronic-11, computer-13)
- amod(products-16, computer-13)
- conj and(electronic-11, building-15)
- amod(products-16, building-15)
- dobj(makes-8, products-16)
- dobj(distributes-10, products-16)

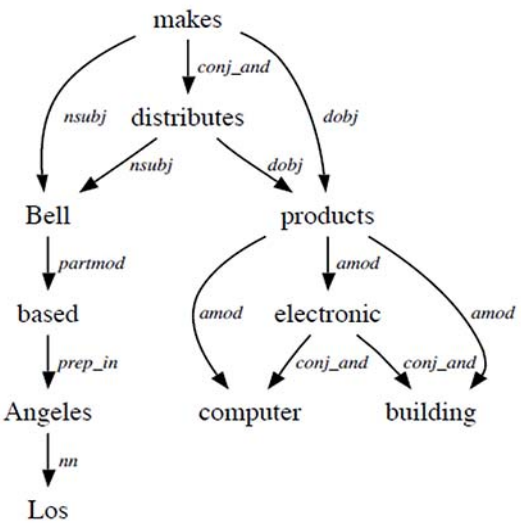

Fig. 1 Sample Dependency Tree

According to the above dependency tree, relationships are as follows

- nsubj – Nominal Subject
- vmod – Verb Modifier
- nn – Noun Compound Modifier
- conj – Conjunction
- prep – Preposition
- amod – Adjectival Modifier
- dobj – Direct Objcet

III. Wordnet Lexical Database

Wordnet is the product of a research project at Princeton University, which attempted to model the lexical knowledge of native speaker of English [8]. In Wordnet nouns, verbs, adverbs and adjectives are organized in to a variety of semantic relations and synonym sets (synsets). Each synset has a gloss that defines the concept of the word. For example, the word car, automobile and motorcar are synsets that represent the concept define by gloss: four wheel Motor vehicle usually propelled by an internal combustion Engine. There are four commonly use semantic relations for nouns [7],

1) Hyponym/hypernym (IS-A/HAS-A)
2) Part meronym/part holonym (PART-OF)
3) Member meronym/member holonym (MEMBER-OF)
4) Substance meronym/substance holonym (Substance-Of)

For an example, apple is a fruit (IS-A) and keyboard is part of computer (PART-OF). However, in the Wordnet, around 80% of relations are based on Hyponym/hypernym.

IV. Measuring the Similarity Between Words

Although the focus is to measure, the sentence similarity at the core level the basic measurement is measuring the word similarity. There are two major types of word similarity measures. Those are,

1. Path Based Measures
2. Information Content Based measures

Based on these two major paths we identified several similarity measures for word similarity. Those are shown in Fig. 2.

A. Path Based Measures

In path-based measures, it considers about the path to a word in the hierarchy of the lexical database. Here in Wordnet we consider about the IS-A hierarchy of the words according to the position they stay in the hierarchy. In the path-based measures, we get the length of the path between the word nodes and use them as a measure of a similarity. Simply the motivation is if the length between two word nodes inversely proportional to the similarity between those two words. In each of the measuring techniques, which follow the path, based measuring approach follows the above concept and handles the path lengths differently.

B. IC Based Measures

In this method's major focus is to measure the similarity of words based on how much of information it holds in the given corpus. In the implementation of information content-based methods, we keep a collection of information in which the data describes how much of information is connected to a certain word.

In this algorithm implementation, we used the Lin's measure, which is based on the information content metrics. In order to support the similarity measure between words we used the WS4J (Wordnet Similarity for Java) [6] similarity measuring library.

V. Sentence Semantic Similarity Measuring Procedure

The similarity measuring procedure is based on four steps as follows.

1) *Pre-processing:* In the preprocessing stage lemmatizing, stemming, POS tag identification and Grammatical Dependencies extraction is done. For the lemmatizing purposes and stemming purposes we used StanfordCoreNLP library's lemmatize capability and the Porter's stemmer stem capability in the SnowBallStemmer library respectively.Also at this stage, we extract the POS tags and the grammatical dependencies from the Stanford dependency parser.

2) *Word Node extraction:* In word node extraction, what we did was extracting the noun nodes and verb nodes. Then also extracts the subordinates of each of the noun node and the verb node. In the similarity measuring process we use the subordinates of the word nodes (adjectives and adverbs) to determine how much similar the two given sentences.

3) *Word vector creation:* In this process, we create the two node bases from all the noun nodes and the verb nodes. Then on these base vectors, we measure the similarity of the word node vectors.

4) *Similarity measure:* In the last step, what we do is calculating the similarities of the word vectors and evaluating the final similarity measuring value of two sentences in both Syntactic measure and Semantic measure. Then calculate the total similarity of two sentences based on these two measures defining weights for each measure.

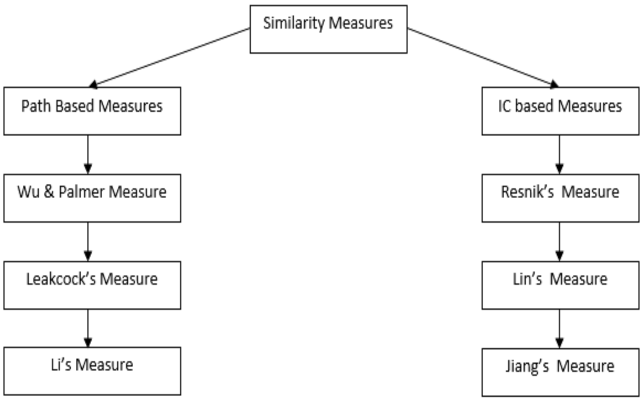

Fig. 2 Word Similarity Measuring Methods

VI. Similarity Measuring Implementation

In the similarity measuring process, the Figure 3 shows the workflow that used for the implementation.

In this process, there are several measuring equations have been used for the similarity measures [1].

$$SimD\ (D1,D2) = \frac{Simd\ (D1,D2) + Simd\ (D2,D1)}{2}$$

Equation 1: Average Subordinates Similarity

$$SimN\ (N1,N2) = \alpha_1 Sim(w_1, w_1) + \alpha_1 SimD\ (D1,D2)$$

Equation 2: Similarity between two Word Nodes

Where $\alpha_1 + \alpha_2 = 1$

In order to generate the word vectors for each of the sentences bellow equations used. We need to create word node vectors for both verbs and the noun nodes.

$$Vv_{1k} = {}_{i=1}^{VN_1}max(Sim_N(VN_{1i}, VNb_k))$$

Equation 3: Verb Feature Item for Sentence one

$$Vv_{2k} = {}_{i=1}^{VN_2}\mathrm{max}\left(Sim_N(VN_{2i}, VNb_k)\right)$$

Equation 4: Verb Feature Item for Sentence two

$$Nv_{1l} = {}_{i=1}^{NN_1}\mathrm{max}\left(Sim_N(NN_{1i}, NNb_l)\right)$$

Equation 5: Noun Feature Item for Sentence one

$$Nv_{2l} = {}_{i=1}^{NN_2}max(Sim_N(NN_{2i}, NNb_l))$$

Equation 6: Noun Feature Item for Sentence two

For measuring the vector similarity,

$$Sim_v(v_1, v_2) = \left(\frac{\sum_{k=1}^{|VN_b|} Vv_{1k} \times Vv_{2k}}{\sqrt{\sum_{k=1}^{|VN_b|} Vv_{1k}^2} \times \sqrt{\sum_{k=1}^{|VN_b|} Vv_{2k}^2}}\right)^{\varepsilon}$$

Equation 7: Total Verb vector Similarity

$$Sim_\eta(\eta_1, \eta_2) = \left(\frac{\sum_{k=1}^{|NN_b|} Nv_{1l} \times Nv_{2l}}{\sqrt{\sum_{l=1}^{|NN_b|} Nv_{1l}^2} \times \sqrt{\sum_{l=1}^{|NN_b|} Nv_{2l}^2}}\right)^{\varepsilon}$$

Equation 8: Total Noun vector Similarity

After all the next task of measuring, the semantic similarity is based on the following equation.

$$Sim_{sem}(S_1, S_2) = \beta_1 \times Sim_v(v_1, v_2) + \beta_2 \times Sim_\eta(\eta_1, \eta_2)$$

Equation 9: Total Semantic Similarity

VII. SENTENCE SYNTACTIC SIMILARITY MEASURE

This method consists with two main steps with usage of binary relationship, which is generated with dependence tree parsing. In here, syntactic information is used to check the similarity of sentence. First, sentence needs to normalized, POS tagged and need to create dependence tree with binary relations. After that, method is created with two main methods

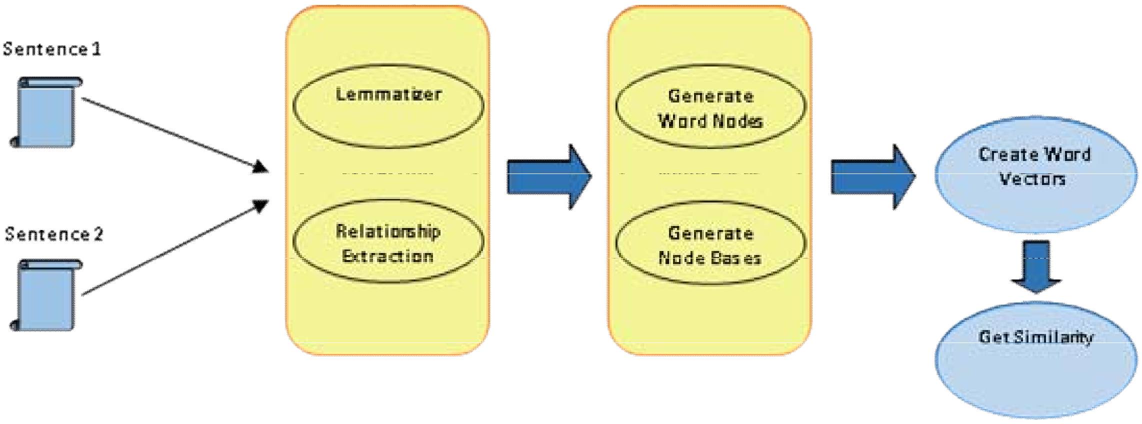

Fig. 3 Algorithm Implementation Workflow

1) Converting Process

Binary relations are extracted from the sentence and uses a convert algorithm to replace parent and child words in binary relation with POS tags, for an example DS (amod[beautiful, girl]) → DS(amod[adj, noun]). Then Ssyn1 and Ssyn2 are extracted from two sentences (S1 and S2).

$$S_{syn1} = \{DS_{syn11}, DS_{syn12}, \dots, DS_{syn1n}\}$$
$$S_{syn2} = \{DS_{syn21}, DS_{syn22}, \dots, DS_{syn2n}\}$$

2) Syntactic similarity measure

After extracting Ssyn1 and Ssyn2 from two sentences, following equation is used to calculate the similarity of two sentences.

$$Sim_{syn}(S_1, S_2) = \frac{2 * |S_{syn1} \cap S_{syn2}|}{|S_{syn1}| + |S_{syn2}|}$$

Equation 10: Total Syntactic similarity

Where, |S_syn1 ∩ S_syn2 |is number of the co-occurring effective syntactic structures in sentence S1 and S2, |S_syn1 |+|S_syn2 |is the total number of the effective syntactic structures in both sentences.

VIII. EXPERIMENTS AND RESULTS

In order to measure the similarity between sentence pairs with the implemented algorithm, the dataset available in [9] was used. There are 65 sentence pairs and here we have used a subset of those sentence pairs. The human ratings of the measurements have been based on 32 human participants and the results of the implemented algorithm are compared based on that.

TABLE 1
SENTENCE SILMILARITY COMPARISON RESULTS

Sentence Pair	Human Rating	Algorithm Result	Error Rate
1	0.16	0.0	1
2	0.09	0.07	0.22
5	0.09	0.19	0.11
9	0.08	0.17	0.12
14	0.48	0.37	0.23
17	0.34	0.32	0.05
21	0.35	0.34	0.02
25	0.4	0.0	1.0
29	0.18	0.19	0.05
30	0.53	0.32	0.39
31	0.46	0.37	0.19
32	0.22	0.28	0.27
37	0.65	0.34	0.47
40	0.78	0.85	0.09
45	0.37	0.27	0.27
47	0.99	0.32	0.67
51	0.68	0.27	0.6
60	1.24	1.57	0.26

62	0.92	0.78	0.15
65	1.31	1.12	0.14

Above table represents how the algorithmic results have been deviated with respect to the human ratings for each of the sentence pairs. As it shows in the above table most of the calculated results shows an error rate which is less than 25% with respect to the actual rating. In order to the sample dataset chosen from [9], approximately 65% of the test results shows less error rate (25% with respect to the actual human rating). When we compare the human ratings and the algorithmic output, most of the time we do not have gained the exact similarity between the sentence pairs. This is because, when a human consider two sentences to determine the similarity they do not use only the semantic or syntactic similarity. Humans also use learned perspectives in determining the similarity, which cannot consider in the algorithmic procedures. Therefore based on the context we need to consider certain heuristics to determine the similarity.

IX. Conclusions

This paper provides better guidance and procedures for measuring the sentence similarity in a well-structured manner. Rather than focusing on sentence similarities of two sentences, this paper provides required background knowledge about "WordNet" lexical database and its usage in the context. Also tin this paper we have provided detailed description about the required fundamental knowledge on word similarity, which is the foundation for the sentence similarity. In order to acquire the word similarity capabilities we have used WS4J library. After providing the required background knowledge, this paper describes the sentence similarity measure in a detailed manner. In addition, we get the help of StanfordCoreNLP and SnowBallStemmer libraries to make thing easier for normalization, POS tagging and the dependence tree parsing. So by adding those descriptions in to this paper to provide a full guideline in usage of the all components we discussed in this paper to make a complete sentence similarity calculation with use of lexical database Wordnet.

Acknowledgement

First of all we would like to thank Department of Computer and Engineering of University of Moratuwa for providing us the required guidance in order to successfully acquire the expected results. Also we would like to thank Dr. Chandana Gamage, the former head of the department and Dr. Malaka Walpola for encouraging us to complete this research.

References

[1] Dingjia LIU, Zequan LIU and Qian DONG, "A Dependency Grammar and WordNet Based Sentence Similarity Measure", Journal of Computational Information Systems 8: 3, 2012.

[2] The Stanford Natural Language Processing Group (2010), *Stanford CoreNLP*,[Online] Available: http://nlp.stanford.edu/software/corenlp.shtml

[3] Hudson, Richard. *An English Word Grammar*. Oxford: Basil Blackwell, 1990.

[4] (2013) WordNet website , [Online]. Available:http://wordnet.princeton.edu/.

[5] Snowball, *Stemmers*, [Online] Available: http://snowball.tartarus.org/algorithms/porter/stemmer.html

[6] WS4J (2013) – "Wordnet Similarity for Java", [online] Available: https://code.google.com/p/ws4j/

[7] Lingling Meng, Runqing Huang and Junzhong Gu, "A Review of Semantic Similarity Measures in WordNet", International Journal of Hybrid Information Technology Vol. 6, No. 1, January, 2013

[8] C. Fellbaum, editor. "WordNet: An Electronic Lexical Database". MIT Press, Cambridge, USA, 1998.

[9] James O'Shea*, Zuhair Bandar, Keeley Crockett, David McLean, "Pilot Short Text Semantic Similarity Benchmark Data Set: Full Listing and Description", manchester Metropolitan University. United Kindom, 2009.