A Study of Using Syntactic Cues in Short-text Similarity Measure

Po-Sen Huang[1,2], Po-Sheng Chiu[3], Jia-Wei Chang[4], Yueh-Min Huang[1], Ming-Che Lee[5]

[1] Department of Engineering Science, National Cheng Kung University, Taiwan
[2] Research Center of Happiness Studies and Technology, Tajen University, Taiwan
[3] Department of E-learning Design and Management, National Chiayi University, Taiwan
[4] Department of Computer Science and Information Engineering,
National Taichung University of Science and Technology, Taiwan
[5] Department of Computer and Communication Engineering, Ming Chun University, Taiwan
phuang@tajen.edu.tw, chiups@mail.ncyu.edu.tw, jiaweichang.gary@gmail.com,
huang@mail.ncku.edu.tw, leemc@mail.mcu.edu.tw

Abstract

Short-text semantic similarity is an essential technique of natural language search and is widely used in social network analysis and opinion mining to find unknown knowledge. Such similarity measures usually measure short texts with 10-20 words. Similar to spoken utterances, short texts do not necessarily follow formal grammatical rules. The limited information contained in short texts and their syntactic and semantic flexibility make similarity measures difficult. Therefore, this study designed and tested a part-of-speech-based short-text similarity algorithm to solve those problems. The effects of evaluating different parts of speech are thoroughly discussed. The proposed algorithm achieved the best performance using word measures corresponding to different parts of speech.

Keywords: Short-text similarity, Semantic analysis, Part-of-speech, WordNet

1 Introduction

Short-text semantic similarity (STSS) usually measures text similarity of 10-20-word-long, or even grammatically incomplete, phrases; short texts and grammatically incomplete sentences are widely used in social networks. STSS has many potential applications for social network analysis (SNA). For example, Chelmis et al. [1] demonstrated that semantic similarity improves the accuracy of predicting the communication intention in social networks. Huang and Yang [2] proposed a semantic-clustering-based method to detect communities. Xu et al. [3] used STSS to develop a system for personalized academic researcher recommendation. STSS can also be applied in databases as an assessment standard to seek unknown information [4]. Furthermore, it can be employed in text categorization [5], recommendation mechanisms [6], machine translation (MT) [7], or sentiment analysis [8].

STSS analysis is closely related to recognizing textual entailment (RTE), applied to many natural language processing (NLP) tasks. To differentiate STSS and RTE, STSS assumes bidirectional graded similarity equivalence between a pair of short texts, whereas RTE uses a directional equivalence. Given the text “A dog is a pet, but a pet is not necessarily a dog,” the result of RTE is a yes or no decision (e.g., a pet is not a dog). STSS is more similar to a graded similarity (e.g., a dog and a pet are more similar than a dog and a computer). Therefore, RTE is an asymmetric work, but STSS is a symmetrical task. The graded bidirectional evaluation is useful for many NLP tasks, such as MT evaluation, information extraction, question-answering systems, and text summarization. The task definitions illustrate the differences between RTE and STSS, but STSS can be a valuable feature of RTE; however, it does not always contain sufficient information [9].

In general, STSS tasks can be divided into corpus- and ontology-based measures using statistics from a large corpus (e.g., British National Corpus or Brown Corpus) or WordNet [10], a lexical database. Despite STSS being studied extensively [11-15], the problem of time complexity has not been given adequate attention in related studies. As the concept of big data attracts attention, time complexity has become a crucial factor of semantic analysis methods in areas such as information retrieval and SNA. Existing semantic analysis approaches have performance limitations caused by polysemy, words having multiple meanings. Because the test for polysemy employs the ambiguous concept of relatedness, judgments of polysemy can be controversial. However, traditional statistical measures suggest that words occurring in the same contexts tend to have similar meanings [16]; thus, they cannot solve

*Corresponding Author: Jia-Wei Chang; E-mail: jiaweichang.gary@gmail.com
DOI: 10.3966/160792642019052003017

the polysemy problem because they only evaluate the relevance between words rather than recognize which word sense is relevant when the words appear in different contexts. STSS performance is weakened using poor word similarities [17].

Fortunately, WordNet has sufficient semantic information and respects the syntactic categories of noun, verb, adjective, and adverb. It is possible to include different parts of speech (POSs) to allow a word to have different meanings. A natural language sentence can be considered a set of words with POSs indicating that each word contains a specific sense of its particular POS. The characteristics can reduce the decrease the adverse effect of word polysemy and extract the latent semantics from the contexts. However, most critical aspect in application is the repeatability of ontology-based measures. Development of grammar parsers (e.g., Stanford Parser [18] and Link Grammar [19]) has allowed syntactic parsing to provide additional cues (e.g., POS, parser trees, or typed dependencies) to reduce the ambiguity of word matching in STSS tasks. The literature [9, 12, 14] also shows that syntactic cues improve measures of short-text similarity.

The goal of this study was to improve the performance of existing STSS and develop an algorithm with low time complexity. We incorporated different WordNet-based word measures to address word pairs with specific POSs. Furthermore, we used the string similarity algorithm from [11] to enhance the evaluation of semantic similarity of unknown words in WordNet. We improved our algorithm through tuning based on both our assumptions and famous large training datasets and validated the performance by using well-known test datasets for STSS and RTE tasks. This study verified the algorithm and our assumptions using new datasets and large corpora with annotated test sets. The proposed algorithm achieved satisfactory performance for both STSS and RTE tasks.

2 Literature Review

Natural language refers to a native representation that changes because of cultural evolution. The changes lead to problems of syntactic and semantic ambiguity in NLP tasks. Therefore, we introduce issues related to word polysemy, existing word similarity, and short-text similarity measures in this section.

2.1 Word Polysemy

A polysemy refers to a word or phrase with multiple meanings. Because the test for polysemy employs the ambiguous concept of relatedness, judgments of polysemy can be controversial. Because applying pre-existing words to new situations is a natural process of language change, looking at etymology is helpful in identifying polysemy. English has many words with polysemous meanings. For example, the verb "to get" can mean "procure" ("I'll get the drinks"), "become" ("she got scared"), "have" ("I've got three dollars"), or even "understand" ("I get it"). To solve the polysemy problem, researchers began studying word sense disambiguation (WSD). Navigli [20] stated, "WSD is the ability to identify the meaning of words in context and is considered as an AI-complete problem, that is, a task whose solution is at least as hard as the most difficult problems in artificial intelligence."

One well-known WSD uses WordNet to utilize knowledge resources to infer the senses of words in context. Knowledge-based methods usually have the advantage of broad coverage because of the use of large-scale knowledge resources [20]. The availability of computational lexicons has allowed the development of many approaches for analyzing and exploiting the structure of the concept network from WordNet. Knowledge-based approaches calculate the gloss overlap between the meanings of two words [21-22]; the senses of two targets whose definitions have the highest overlap are assumed to be the correct ones. Distance-based measures [23-24] develop a similarity measure based on the distance of two senses and focus on hypernym links and scale the path length with the overall depth of the taxonomy. Information content (IC)-based measures [25-27] use the notion of IC shared by words in context. IC measures determine the specificity of the concept that subsumes the words in the taxonomy based on the idea that more specific concepts subsume more words if they are assumed to be more closely related semantically. To solve the polysemy problem, we expect WordNet-based word similarity based on WSD techniques to play a role in measuring semantic similarity.

2.2 Word Similarity Measures

Han [17] said, "Using a poor similarity measure could reduce the performance of the proposed method because word similarity is the basic unit of STSS work." The crucial element of STSS is to measure the word similarity between concepts. In general, there are statistical (or corpus-based) and ontology-based (or WordNet-based) measures, which are described as follows.

Corpus-based word similarity measures [28-30] depending on the occurrence of a word and co-occurrence of two words from a large high-quality corpus. The advantage of a statistical approach is to measure novel words by adding new articles to the corpus, but this approach cannot resolve the problem of word polysemy, because statistical word similarity represents the relevance between words rather than identifying which word sense provides the relevance when the words appear in different contexts. For instance, word "saving" could be a noun ("she lived on her savings"), an adjective ("old-fashioned housewives were usually good at saving"), or a verb ("we've been saving for five years to buy a house"). Corpus-based measures usually cannot choose the POS or recognize the proper sense of "saving" when that word is used with "money" (noun), "keep" (verb), or "luxurious" (adjective). In addition, there are problems with corpus-based methods. (1) Adding a new article to the corpus is difficult because it should be evaluated for quality by experts. (2) Statistical approaches often do not work with large high-quality corpora because resources are few and old. (3) The repeatability of performance is a concern when using different corpora or adding new articles of unknown quality.

The possible solution to the polysemy problem is dictionary-based methods, which are based on stable and strict ontology. WordNet was developed by the Cognitive Science Laboratory at Princeton University in the 1990s. Nouns, verbs, adjectives, and adverbs are grouped into cognitive synonyms called synsets, and each synonym expresses a distinct concept. As an ordinary online dictionary, it lists subjects alphabetically along with explanations. In addition, it shows semantic relations among words and concepts. The latest version is 3.0, which contains more than 150,000 words and 110,000 synsets. In WordNet, the lexicalized synsets of nouns and verbs are organized hierarchically through hypernymy and hyponymy. These characteristics can reduce the ambiguity of words. The three types of WordNet-based measures are presented (detailed explanation in [20, 31]).

1. Distance-based measures: By using the hierarchical structure of WordNet (or any other taxonomy with a similar structure), the path length between concepts can be used to measure their similarity Three measures of this type are PATH, WUP [23], and LCH [24].

2. IC-based measures: Also using a hierarchical structure, the specificity of a concept is higher for more specific concepts. Three measures of this type are RES [25], LIN [26], and JCN [27].

3. Gloss-based measures: These measures use the glosses associated with each concept in WordNet. Well-known measures of this type are VECTOR [21] and LESK [22].

However, distance- and IC-based measures greatly depend on the hierarchical structure, which is only available for nouns and verbs (i.e., unavailable for adjectives and adverbs) [17]; this weakness means that these two measures cannot solve the polysemy problem well for adjectives and adverbs. However, we expect gloss-based measures to address pairs adjectives and adverbs. The practical superiority of WordNet-based measures is based on the stable repeatability of performance ensured by strict ontology.

2.3 Short-text/Sentence Similarity Measures

In general, there are three types of unsupervised STSS tasks: corpus-based, ontology-based, and hybrid approaches. Islam's STS [11] is a corpus-based method that removes stop words and builds an m × n similarity matrix of meaningful words from two short texts. It uses statistical word similarity and string similarity to compute the m × n word pairs of the matrix, sums maximum-valued matrix elements, and multiplies the sum by the reciprocal harmonic mean of m and n to obtain a balanced similarity score between 0 and 1. The most significant difference of STS is the use of the longest common subsequence (LCS) to design the string-matching algorithm, which can evaluate proper nouns and improve the word similarity measurement for words with less statistical information. Although STS is an excellent method, it has some problems. First, it is a pair-matching method with high time complexity to compute the meaningful words for each word pair in the similarity matrix. Second, it uses corpus-based word similarity alone and cannot solve the polysemy problem.

Tsatsaronis's Omiotis [12] is an algorithm to compute text relatedness based on WordNet and uses POSs and various semantic relations (i.e., synonymy, antonymy, hypernymy, hyponymy, holonymy, meronymy, and metonymy) between words to perform WSD and obtain word similarities. Omiotis improves STSS evaluation by reducing the ambiguity of a word pair and expanding the semantic relations of the word pair. However, this algorithm has high time complexity because it uses many semantic relations and redundant matching processes.

Oliva's SyMSS [9] also uses WordNet-based word measures and syntactic cues, including the parse tree, to perform WSD and evaluate STSS. It obtains the parse trees of two sentences or short texts by using the grammar parser and uses the structure of two parse trees to compute word similarities when the two words perform the same syntactic role in the syntactic structures. The novel idea of SyMSS is considering the syntactic information of short texts and assigning weights to different syntactic roles. It employs syntactic information to perform WSD and reduce word matching, which gives SyMSS lower time complexity. However, using the structure of the parse tree to match words seems inappropriate because it can lead to inaccurate results when two short texts have the same meaning but different syntactic structures.

Li's STASIS [15] employs a semantic vector space formed by a union of the words in two sentences; it combines WordNet- and corpus-based word similarities to compute two semantic vectors by matching with the vector space. STASIS evaluates two vectors with a vector space model (VSM) to obtain a semantic similarity between sentences and depends on the syntactic rule (i.e., word order) to design a similarity measure. STASIS combines the VSM, semantic similarity, and word order method for sentence similarity measurement. However, STASIS does not remove the stop and meaningless words that can lead to inaccurate results. It also possesses high time complexity because it depends on the VSM. These problems make STASIS impractical in real life situations.

To summarize, Omiotis and SyMSS use syntactic information, POSs, and parse trees to reduce the ambiguity between words and match words with the same syntactic roles. Discovering syntactic information not only helps STSS measurement by reducing the adverse effects of word polysemy but also improves efficiency. It is the proper direction to obtain information useful for approaches to NLP and semantic analysis.

3 Methodology

This section describes the proposed similarity algorithm in detail. Our approach obtains similarity from semantic and syntactic information present in the compared natural language sentences. A natural language sentence can be defined as a set of words with POSs, rather than individual word strings, and each word with a specific POS contains a particular meaning and plays a certain role in the sentence. Based on this idea, the proposed method is expected to reduce the ambiguity of short texts and extract the latent semantics from syntactic clues. We employ different word metrics for measuring different POSs to improve performance and mitigate the limitations of WordNet.

3.1 Framework and Core Functions

The proposed framework is divided into two subsystems: semantic analysis and semantic evaluation. A normal user or software agent enters two sentences; the output is a similarity score of the two sentences. The semantic analysis subsystem first formalizes the input sentences into tokens and builds the structure of the semantic matrix according to POSs. The semantic evaluation subsystem evaluates the similarity of each POS using WordNet measures and extracts the maximum semantic joint set of the matrix.

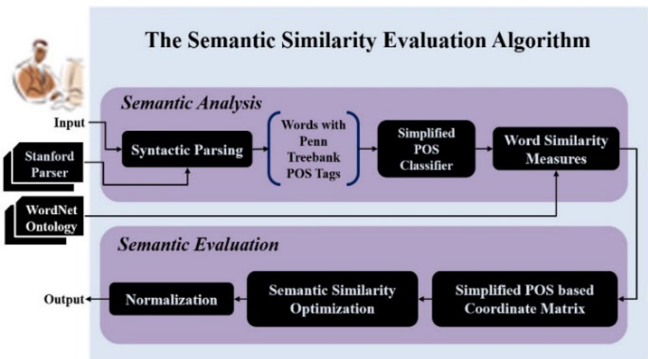

Figure 1. Framework of short-texts semantic similarity algorithm

3.2 Word Similarity Measures

In the proposed method, word similarity is calculated by WordNet-based semantic measurements and our string similarity algorithm as in Formula (1). According to different POSs, it uses corresponding measurements to obtain word similarity between W_1 and W_2. For WordNetSimilarity, all words are necessary for word stemming that helps to measure semantic similarity; however, StringSimilarity does not need this. Formula (1) shows the detailed methods of WordNet for three conditions. (1) If the two words are nouns or verbs, semantic similarity is measured based on distance or IC. (2) If the two words are adjectives or adverbs, gloss-based metrics are applied. (3) If the two words are other POSs, the optional string similarity measure is used because WordNet cannot evaluate them.

$$WordSimilarity(w_1, w_2, pos) = \begin{cases} WordNetSimilary_{Distance\|IC}(w_1, w_2), \textbf{\textit{if}}\ pos\ \textbf{equal to}\ Noun \| Verb \\ WordNetSimilary_{Gloss}(w_1, w_2), \textbf{\textit{if}}\ pos\ \textbf{equal to}\ Adj. \| Adv. \\ StringSimilarity(w_1, w_2), \textbf{\textit{if}}\ pos\ \textbf{equal to}\ Other \end{cases} \quad \textbf{(1)}$$

There are three bases for methods employed by WordNet: distance, IC, and gloss. However, distance- and IC-based methods depend on the hierarchical relationship, strongly supported only for nouns and verbs. More reliable semantic similarity of adjectives and adverbs can be evaluated through gloss-based metrics (detailed explanation in Section 2.2); therefore, VECTOR [21] was employed in this study. However, a method for measuring semantic similarity when the two words are other POSs is required because WordNet only has nouns, verbs, adjectives, and adverbs. To solve this problem, the string similarity algorithm [11] is used in the proposed short-text similarity measure. The string similarity measure is based on lexical matching of words or parts of words. It calculates string similarity using three modifications of the LCS algorithm.

3.3 POS-based Semantic Measure for Short Texts

The proposed algorithm determines the similarity of two natural language sentences based on the POS information and word similarity measurement processes, which can be separated into three functions: simplified POS classification, semantic similarity optimization (SSO), and semantic similarity normalization (SSN).

The first step is syntactic parsing and POS simplification. The Penn Treebank POS tags must be simplified because WordNet only contains nouns, verbs, adjectives, and adverbs. We propose simplified rules listed in Table 1, which presents the simplified POS tagset in this approach. The first column shows the simplified tags that can help us obtain superior performance and lower complexity. The second column shows the POS tags of the Penn Treebank [32] mapped to each simplified tag.

Table 1. Simplified POS tagset

Simplified POS	Penn Treebank POS
Noun (n)	*NN, NNS, NNP, NNPS*
Verb (v)	*VB, VBD, VBG, VBN, VBP, VBZ*
Adjective (a)	*JJ, JJR, JJS*
Adverb (r)	*RB, RBR, RBS*
Others (o)	*CC, CD, DT, EX, FW, IN, LS, MD, PDT, POS, PRP, PRP$, RP, SYM, TO, UH, WDT, WP, WP$, WRB*

Algorithm P_1 accepts a sentence S and a lookup table of simplified tagset η, then invokes the syntactic parsing function to generate the Penn Treebank POS and returns the set of simplified POSs, as shown in Figure 2. This crucial preprocessing is based on the same simplified POSs to improve semantic similarity between words and word matching. Words with POSs from a pair of sentences can form a matrix, named the POS-based coordinate matrix (PCM). In Figure 2 and Algorithm P_2, a PCM was composed of words with the same POSs from two texts.

Algorithm P_1. Simplified POS Classifier (SPC)

```
INPUT: SENT, η /* SENT is the input sentence, and
        η is a lookup table as Table 1*/
OUTPUT: SimplifiedPOS_SENT
PennPOS_SENT ← Stanford_Parser(S)
FOR ALL T_i ∈ PennPOS_SENT
DO
SimplifiedPOS_SENT ← LookupSimplifiedTag
(η, T_i)
END FOR
RETURN SimplifiedPOS_SENT
```

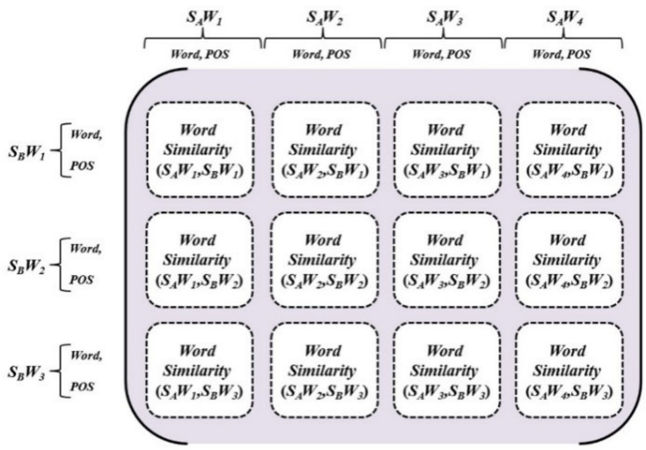

Figure 2. Diagram of POS based coordinate matrix

After preprocessing phrase P_1, we set the POS set with fewer words as the columns and the other as the rows. For each row, the maximal term is reserved and forms a POS vector (PV), which represents the maximal semantic inclusion of a specific POS between two sentences. Figure 2 illustrates the structure of the PCM and PVs; each word pair in the PCM is composed of the words with POSs from S_A and S_B.

Each PCM represents a correlation of certain words because there may exist similar syntactic roles in sentences, for which the corresponding PV quantifies the semantic information and extracts semantics from these words. Figure 3 illustrates the process of *SSO*. In the example, S_AW_1-S_BW_1 and S_AW_3-S_BW_1 are word pairs with the same POSs (i.e., nouns), and S_AW_3-S_BW_1 denotes the maximum word similarity (MWS) of nouns for S_A and S_B. The MWSs of nouns, verbs, adjectives, and adverbs are obtained through the same process. According to Algorithm P_2, we sum all the MWSs to obtain the optimized result, MWS_{SUM}.

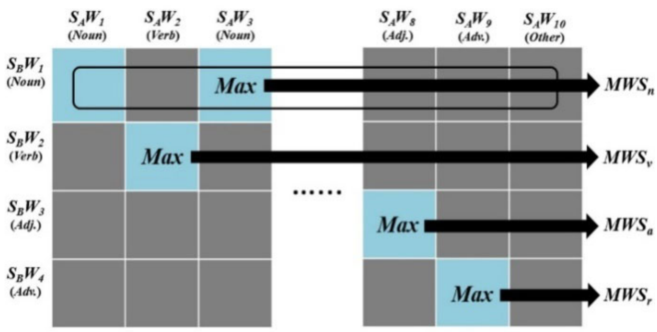

Figure 3. Diagram of semantic similarity optimization

Algorithm P_2. Semantic Similarity Optimization (SSO)

```
INPUT: SimplifiedPOS_A,SimplifiedPOS_B /*
        Simplified POS sets of S_A, S_B */
OUTPUT: MWS_SUM /* The Sum of Maximum Word
        Similarity of S_A, S_B */
ROW ← MAX (SimplifiedPOS_A, SimplifiedPOS_B)
COL← MIN(SimplifiedPOS_A,SimplifiedPOS_B)
FOR ALL c_x ∈ COL DO
FOR ALL r_y ∈ ROW DO
IF c_x.pos EQUAL r_y.pos THEN
PV[x]←MAX(PV[x], WordSimilarity
(c_x.w, r_y.w, pos))
END IF
END FOR
END FOR
FOR 0 TO |COL|
MWS_SUM ←MWS_SUM + PV[x]
END FOR
RETURN MWS_SUM
```

After optimizing, we normalize the MWS_{SUM} for our algorithm to provide a similarity score between 0 and 1, inclusively. Algorithm P_3 shows the processes of normalization and returns a normalized coefficient (NC) for two short texts. SSN is based on the reciprocal harmonic mean and the lengths of two short texts, which can help our method reach a balanced similarity score. Finally, Algorithm P_4 details all evaluation processes.

Algorithm P_3. Semantic Similarity Normalization (SSN)

INPUT: S_A,S_B /* The original text of S_A, S_B*/

OUTPUT: NC /* Normalized Coefficient of S_A, S_B*/

1. $\text{Length}_1 \leftarrow$ Counting_Words(S_A)
2. $\text{Length}_2 \leftarrow$ Counting_Words(S_B)
3. NC $\leftarrow$ (Length_1 + Length_2) / (2*Length_1*Length_2)
4. **RETURN**NC

Algorithm P_4. Semantic Similarity Algorithm

INPUT: S_A, S_B, η /* Raw sentencesA, B and η is Table 1*/

OUTPUT: STSS_{AB} /* Short-text semantic similarity between S_A, S_B*/

1. $\text{SimplifiedPOS}_A \leftarrow$ SPC (S_A, η)
2. $\text{SimplifiedPOS}_B \leftarrow$ SPC (S_B, η)
   /* SPC is the Algorithm P_1 which can get the information of simplified POS. */
3. $\text{MWS}_{\text{SUM}} \leftarrow$ SSO (SimplifiedPOS_A, SimplifiedPOS_B)
   /* SSO is the Algorithm P_2which returns the sum of maximum word similarity between S_A and S_B.*/
4. NC$\leftarrow$ SSN (S_A, S_B)
   /* SSN is the Algorithm P_3 which returns the normalized coefficient between S_A and S_B.*/
5. $\text{STSS}_{AB} \leftarrow \text{MSW}_{\text{SUM}}$ *NC
6. **RETURN**STSS_{AB}

3.4 Walkthrough with Examples

This section gives an example to demonstrate the proposed similarity algorithm. Let A = "A cemetery is a place where dead people's bodies or their ashes are buried." Let B = "A graveyard is an area of land, sometimes near a church, where dead people are buried." Finally, let C = "Your signature is your name, written in your own characteristic way, often at the end of a document to indicate that you wrote the document or that you agree with what it says." The examples are introduced in more detail in the following section. In this example, we compare the semantic similarities between A and B, A and C, and B and C. Algorithm P_1 first generates the corresponding POSs for each sentence; the results are shown in Table 2. After preprocessing, the compared sentence pair is sent to Algorithm P_2, which produces word pairs according to their common POSs, and forms PCMs.

Figure 4 show the PCMs and their word-to-word similarities for A and B. There are five PCMs in pair A-B. The first PCM is a noun PCM, a 5 × 5 matrix with nouns from S_A and S_B. The verb PCM is a 3 × 3 matrix with verbs from A and B. The adjective PCM is a 1 × 1 matrix formed by the adjectives of A and B. The adverb PCM is a 1 × 2 matrix of adverbs. The "others" PCM is a 6 × 8 matrix with the words of other POSs comprising

Table 2. Examples with simplified POS

Sentences	Raw sentences with simplified POS
SENT_A	A[*o*] cemetery[*n*] is[*v*] a[*o*] place[*n*] where[*r*] dead[*a*] people[*n*] 's[*o*] bodies[*n*] or[*o*] their[*o*] ashes[*n*] are[*v*] buried[*v*].[*o*]
SENT_B	A[*o*] graveyard[*n*] is[*v*] an[*o*] area[*n*] of[*o*] land[*n*], [*o*] sometimes[*r*] near[*o*] a[*o*] church[*n*], [*o*] where[*r*] dead[*a*] people[*n*] are[*v*] buried[*v*].[*o*]
SENT_C	Your[*o*] signature[*n*] is[*v*] your[*o*] name[*n*], [*o*] written[*v*] in[*o*] your[*o*] own[*a*] characteristic[*a*] way[*n*], [*o*] often[*r*] at[*o*] the[*o*] end[*n*] of[*o*] a[*o*] document[*n*] to[*o*] indicate[*v*] that[*o*] you[*o*] wrote[*v*] the[*o*] document[*n*] or[*o*] that[*o*] you[*o*] agree[*v*] with[*o*] what[*o*] it[*o*] says[*v*].[*o*]

NOUN

S_A/S_B	graveyard	area	land	church	people	*MWS*
cemetery	*1*	~~0.17~~	~~0.17~~	~~0.08~~	~~0.08~~	**1**
place	~~0.2~~	*0.5*	~~0.33~~	~~0.14~~	~~0.17~~	**0.5**
people	~~0.08~~	~~0.13~~	~~0.5~~	~~0.2~~	*1*	**1**
bodies	~~0.1~~	~~0.33~~	~~0.2~~	*0.5*	~~0.25~~	**0.5**
ashes	~~0.09~~	~~0.13~~	*0.17*	~~0.1~~	~~0.13~~	**0.17**

VERB

S_A/S_B	is	are	buried	*MWS*
is	*1*	*1*	~~0.33~~	**1**
are	*1*	*1*	~~0.33~~	**1**
buried	~~0.33~~	~~0.33~~	*1*	**1**

ADJECTIVE

S_A/S_B	dead	*MWS*
dead	*1*	**1**

ADVERB

S_A/S_B	sometimes	where	*MWS*
where	~~0~~	*1*	**1**

OTHERS POS

S_A/S_B	A	an	of	,	near	a	,	.	*MWS*
A	*1*	~~0.33~~	~~0~~	~~0~~	~~0.08~~	*1*	~~0~~	~~0~~	**1**
a	*1*	~~0.33~~	~~0~~	~~0~~	~~0.08~~	*1*	~~0~~	~~0~~	**1**
's	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	*0*	**0**
or	~~0~~	~~0~~	*0.17*	~~0~~	~~0.08~~	~~0~~	~~0~~	~~0~~	**0.17**
their	~~0~~	~~0~~	~~0~~	~~0~~	*0.03*	~~0~~	~~0~~	~~0~~	**0.03**
.	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	*1*	**1**

Figure 4. PCM of sentences A, B of walk-through

A and B. In step 6, Algorithm P_2 evaluates single-word similarity by using the WordNet measures and the string similarity method. The PCMs of pairs A-C and B-C follow the same process. This phase evaluates all possible semantics between similar syntactic roles; in general, a word may be matched once or more. The next phase reduces each PCM to a PV using SSO and reserves the maximal value of each row. In the pair A-B, PV = {1, 0.5, 1, 0.5, 0.17, 1, 1, 1, 1, 1, 1, 1, 0, 0.17, 0.03, 1}.

In the pair A-C, PV = {0.11, 0.5, 0.33, 0.33, 0.14, 1, 1, 0.33, 0.09, 0, 1, 1, 0, 1, 0.4, 1}, and in the pair B-C, PV = {0.11, 0.25, 0.2, 0.17, 0.33, 1, 1, 0.33, 0.09, 0.03, 0, 1, 0.33, 1, 1, 0.08, 1, 1, 1}. Then, all MWSs of the PVs are summed to obtain the MWS_{SUM} of pairs A-B, A-C, and B-C. In the normalization procedure, following Algorithm P_3 uses the number of elements of two sentences to compute the NC of the sentence pair. In the examples, we obtain $NC_{A\text{-}B}$ = 0.05756579, $NC_{A\text{-}C}$ = 0.045138888 and $NC_{B\text{-}C}$ = 0.040204678. Finally, MWS_{SUM} multiplied by NC equals the final scores: A-B = 0.65, A-C = 0.37, and B-C = 0.40. Pair A-B has the highest similarity score (0.65); therefore, the proposed method can satisfactorily evaluate semantics.

4 Experiments

In this section, we present our verification of the algorithm design. Training data from STSS and RTE tasks were used to improve the performance of the proposed method.

4.1 Datasets of STSS

To explore the effects of using different POS sets, the training dataset used was from SemEval-2012 [33]. The two test datasets from Li et al. [15] and SemEval-2012 were used to support empirical evidences for the proposed algorithm and compare the supervised and unsupervised approaches with the similarity scores given by human evaluators.

Table 3. Semeval-2012 training data

Data Sets	MSRpar	MSRvid	SMTeur
Source	Microsoft Paraphrase Corpus	Microsoft Video Description Paraphrase Corpus	ACL Workshops on SMT
Domain	News	Video Descriptions	European Parliament Proceedings
Length in Terms (avg.)	6-38 (21)	3-25 (8)	3-78 (24)
Number of Pairs	750	750	734

SemEval-2012. In SemEval-2012 [33], the training data contained 2234 sentence pairs from existing paraphrase datasets (MSRpar, MSRvid) and MT evaluation resources (SMTeur). The test data contained 3108 sentence pairs, comprising 1500 sentences pairs for MSRpar and MSRvid, two additional datasets with 399 pairs from SMTeur and 459 pairs from SMTnews, and 750 pairs from a lexical resource (OnWN). The similarity of sentence pairs was rated on a 0-5 scale by human judges with high Pearson correlation scores (approximately 90%).

Li's benchmark. By using the semantic and syntactic information contributing to the understanding of natural language sentences, Li et al. [15] created a similarity measure that is a linear combination based on the similarity of semantic vectors and word order. A preliminary dataset was constructed by Li et al. with human similarity scores provided by 32 volunteers who are all native speakers of English. Li's dataset uses 65 word pairs, which were originally provided by Rubenstein and Goodenough [34] and replaced with definitions from the Collins COBUILD dictionary [35]. The Collins COBUILD dictionary is constructed from a large corpus containing more than 400 million words. Each pair is rated on a scale of 0.0-4.0, according to their similarity of meaning. We used a subset of the 65 pairs to obtain a more even distribution across the similarity range. This subset contained 30 of the original 65 pairs: 10 each with scores of 3-4, 1-3, and 0-1.

4.2 Datasets of RTE

To evaluate the performance of the proposed method in a more challenging application, RTE, we used a larger dataset, the Microsoft Research Paraphrase Corpus (MSRpar) [36]. This dataset consists of 5801 pairs of sentences, including 4076 training and 1725 test pairs, collected from thousands of news sources on the web. Each pair was examined by two human judges to determine whether the sentences in a pair were semantically equivalent paraphrases. Interjudge agreement between annotators was approximately 83%. In this experiment, we used different similarity thresholds from 0-1, with an interval of 0.1, to determine whether a sentence pair was a paraphrase. The performance indicators are defined as follows:

$$\text{Precision} = TP / (TP + FP)$$
$$\text{Recall} = TP / (TP + FN)$$
$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN)$$

Table 4. Microsoft research paraphrase corpus

Data Sets	Category	Number of Pairs	Total Pairs
Training Data	0	1,323	4,076
	1	2,753	
Test Data	0	578	1,725
	1	1,147	

TP stands for true positive, pairs correctly labeled as paraphrases. TN stands for true negative, pairs correctly labeled as not paraphrases. FP stands for false positive, pairs incorrectly labeled as paraphrases. Finally, FN stands for false negative, pairs incorrectly labeled as not paraphrases.

4.3 Training Process

For the STSS task, we used the training data from SemEval-2012, which contained 2234 sentence pairs from MSRpar, MSRvid, and SMTeur (Table 3). To further discuss the influence of different POS sets on STSS, the three POS sets were tuned in training. Table 5 shows that the best performance was achieved using the third POS set (noun, verb, adjective, adverb, and others) and the PATH method. It obtained Pearson correlations of 0.544 for MSRpar, 0.671 for MSRvid, 0.651 for SMTeur. Furthermore, distance-based methods (PATH, LCH, WUP) nearly outperformed IC-based (RES, LIN, JCN) measures.

For RTE, the training dataset of MSRpar was used. In the RTE training results, the proposed measure using all POSs exhibited a stable and reasonable threshold of 0.6. As shown in Table 6, the highest performance (accuracy) was achieved using the third POS set (noun, verb, adjective, adverb, others) and the PATH measure. These results are similar to those of the STSS task. The valuable findings of the training results were used to test the performance of the proposed

Table 5. Pearson correlations of the proposed algorithm with different POS sets on SemEval-2012 Training Data

POS Sets	Methods	MSRpar	MSRvid	SMTeur
[noun, verb]	**PATH**	**.47**	**.442**	**.636**
	LCH	.426	.407	.473
	WUP	.432	.2	.574
	RES	.405	.353	.451
	LIN	.405	.376	.443
	JCN	.405	.376	.443
[noun, verb, adj., adv.]	**PATH**	**.509**	**.648**	**.641**
	LCH	.476	.55	.518
	WUP	.479	.412	.596
	RES	.463	.483	.5
	LIN	.461	.501	.493
	JCN	.461	.502	.494
[noun, verb, adj., adv., others]	**PATH**	**.544**	**.671**	**.651**
	LCH	.514	.556	.589
	WUP	.54	.453	.622
	RES	.504	.488	.58
	LIN	.502	.505	.577
	JCN	.502	.505	.577

Table 6. Results of the proposed algorithm with different POS sets on MSRpar Training Data

POS Sets		Best Thres.	Prec.	Rec.	Acc.
[n, v]	**PATH**	**0.6**	70.83	92.61	**69.24**
	LCH	0.6	70.26	91.14	68.00
	WUP	0.7	68.03	99.00	67.77
	RES	0.5	75.63	76.64	67.62
	LIN	0.5	76.22	74.90	67.23
	JCN	.5	70.69	88.54	67.51
[n, v, a, r]	**PATH**	**0.6**	74.54	88.65	**71.81**
	LCH	0.6	74.00	88.61	71.31
	WUP	0.6	71.77	88.66	68.90
	RES	0.5	71.84	89.41	69.11
	LIN	0.5	72.13	88.34	69.08
	JCN	0.5	72.11	88.31	69.06
[n, v, a, r, o]	**PATH**	**0.6**	77.24	86.23	**73.45**
	LCH	0.6	75.71	87.5	72.54
	WUP	0.6	74.75	85.43	70.75
	RES	0.6	74.86	86.67	71.43
	LIN	0.6	75.23	85.23	71.04
	JCN	0.6	75.18	85.21	71.01

algorithm. Based on the training results, the proposed algorithm achieved the highest performance using all POSs and the PATH measure on both the STSS and RTE tasks.

4.4 Performance Test

The test datasets of SemEval-2012 and MSRpar were used to evaluate the performance of the trained algorithm. SemEval-2012 was for STSS, and MSRpar was for RTE. The experiment examined the consistency of results between the training and the test datasets, specifically, which word measure and POS set result in the best performance.

Table 7. Pearson correlations of the proposed algorithm with different POS sets on SemEval-2012 Test Data

[n, v]	**PATH**	LCH	WUP	RES	LIN	JCN
MSRpar	**.465**	.451	.418	.433	.431	.431
MSRvid	**.733**	.688	.483	.593	.611	.609
SMTeur	**.516**	.442	.429	.431	.426	.426
SMTnews	**.656**	.661	.549	.655	.655	.655
OnWN	**.466**	.449	.404	.423	.418	.417
[n, v, a, r]	**PATH**	LCH	WUP	RES	LIN	JCN
MSRpar	**.562**	.54	.548	.513	.509	.508
MSRvid	**.74**	.663	.457	.569	.581	.577
SMTeur	**.562**	.534	.496	.556	.541	.539
SMTnews	**.659**	.651	.589	.633	.631	.635
OnWN	**.52**	.512	.489	.503	.498	.499
[n, v, a, r, o]	**PATH**	LCH	WUP	RES	LIN	JCN
MSRpar	**.581**	.571	.568	.533	.531	.53
MSRvid	**.76**	.688	.473	.573	.596	.596
SMTeur	**.582**	.559	.486	.576	.575	.575
SMTnews	**.666**	.654	.599	.645	.645	.645
OnWN	**.526**	.521	.504	.518	.517	.517

The proposed method using all POSs and PATH reached the highest Pearson correlations of 0.581 on MSRpar, 0.76 on MSRvid, 0.582 on SMTeur, 0.666 on SMTnews, and 0.526 on OnWN for the STSS task. For RTE, it achieved the highest accuracy, 72.75%, in the MSRpar using all POSs and PATH. Overall, the test datasets exhibited similar results to the training datasets. (1) Nouns and verbs represented the most semantics for natural language. (2) Considering other POSs can improve short-text similarity measures. (3) The proposed method achieved the highest performance using all POSs. (4) The proposed method achieved the highest performance using PATH for nouns and verbs, VECTOR for adjectives and adverbs, and the string measure for other POSs.

Table 8. RTE performance of the proposed algorithm with different POS sets on MSRpar Test Data

POS Sets		Best Thres.	Prec.	Rec.	Acc.
[n, v]	**PATH**	**0.6**	**71.51**	**93.64**	**71.05**
	LCH	0.6	69.10	98.05	69.63
	WUP	0.6	68.00	98.90	68.31
	RES	0.5	69.31	97.88	69.69
	LIN	0.5	69.23	97.71	69.60
	JCN	0.5	69.21	97.65	69.61
[n, v, a, r]	**PATH**	**0.6**	**74.81**	**87.49**	**72.14**
	LCH	0.6	74.77	86.54	71.74
	WUP	0.6	70.21	95.88	70.11
	RES	0.6	72.39	92.49	71.63
	LIN	0.6	71.69	93.51	71.24
	JCN	0.6	71.75	93.48	71.24
[n, v, a, r, o]	**PATH**	**0.6**	**73.86**	**91.37**	**72.75**
	LCH	0.6	74.81	87.53	72.12
	WUP	0.6	71.07	94.25	70.67
	RES	0.6	74.79	87.18	71.94
	LIN	0.6	74.81	86.49	71.65
	JCN	0.6	74.81	86.49	71.65

4.5 Comparisons with other Approaches

The most satisfactory performance measure was PATH (r = 0.83), and this result is consistent with that of Oliva et al. [9]. Table 9 shows the human similarity scores along with those of Li et al. [15], LSA [13], STS Meth. [11], SyMSS [9], Omiotis [12], and the proposed algorithm.

Table 9. Correlationcomparison of STSS task with other approaches on Li's benchmark

No.	Hum	Li	LSA	STS Meth	SyMSS	Omiot.	**Ours (PATH)**
1	.01	.33	.51	.06	.32	.11	.3
5	.01	.29	.53	.11	.28	.1	.4
9	.01	.21	.51	.07	.27	.1	.36
13	.1	.53	.53	.16	.27	.3	.5
17	.13	.36	.58	.26	.42	.3	.35
21	.04	.51	.53	.16	.37	.24	.4
25	.07	.55	.6	.33	.53	.3	.4
29	.01	.34	.51	.12	.31	.11	.42
33	.15	.59	.81	.29	.43	.49	.53
37	.13	.44	.58	.2	.23	.11	.38
41	.28	.43	.58	.09	.38	.11	.41
47	.35	.72	.72	.3	.24	.22	.48
48	.36	.64	.62	.34	.42	.53	.51
49	.29	.74	.54	.15	.39	.57	.55
50	.47	.69	.68	.49	.35	.55	.44
51	.14	.65	.73	.28	.31	.52	.44
52	.49	.49	.7	.32	.54	.6	.51
53	.48	.39	.83	.44	.52	.5	.56
54	.36	.52	.61	.41	.33	.43	.52
55	.41	.55	.7	.19	.33	.43	.44
56	.59	.76	.78	.47	.43	.93	.55
57	.63	.7	.75	.26	.5	.61	.48
58	.59	.75	.83	.51	.64	.74	.57
59	.86	1	1	.94	1	1	.94
60	.58	.66	.83	.6	.63	.93	.58
61	.52	.66	.63	.29	.39	.35	.55
62	.77	.73	.74	.51	.75	.73	.59
63	.59	.64	.87	.52	.78	.79	.56
64	.96	1	1	.93	1	.93	.95
65	.65	.83	.86	.65	.36	.82	.68
(r)	-	.81	.84	.85	.76	.86	**.83**

Our proposed algorithm using PATH achieved a high Pearson correlation coefficient of 0.83. The measure proposed by Li et al. achieved that of 0.82. LSA achieved 0.84. SyMSS achieved 0.76, and Omiotis achieved 0.86. The upper bounds obtained by Li et al. (0.82) and our method (0.83) are the closest to the real upper bound (0.825). Thus, our algorithm employing PATH achieved excellent performance. In brief, our approach identified and quantified latent semantic relationships among syntaxes and words. Our idea yielded satisfactory results for the STSS task.

For RTE, this experiment compared the performance in several categories using the test dataset from MSRpar: (1) two baselines, a random selection and a VSM-cosine-based measure with TF-IDF weighting; (2) corpus-based approaches, PMI-IR [37], LSA [13], STS Meth. [11]; (3) lexicon-based approaches, including that of Mihalcea et al. [38], SyMSS (JCN and VECTOR) [9], Omiotis [12], and LG [14]; (4) machine-learning-based approaches, including those of Wan et al. [39], Zhang and Patrick [40], and Qiu et al. [41], whose is a SVM approach [42].

Table 10 shows that the proposed algorithm outperformed most compared methods. However, it was not superior to the machine-learning-based method proposed by Wan et al. [39]. Overall, our algorithm is an excellent method with a threshold of 0.6, which is a reasonable range to determine whether a sentence pair is a paraphrase. In addition, our algorithm using the six word measures displayed the most satisfactory performance at a threshold of 0.6. This means that our P-STSS is a stable algorithm, regardless of which WordNet-based word measure is used. The proposed approach had the most satisfactory result under the same conditions.

Table 10. Comparisons of RTE task with other approaches on MSRpar

Category	Metric	Best Threshold	Precision	Recall	Accuracy
Corpus-based	PMI-IR	-	70.20	95.20	69.90
	LSA	-	69.70	95.20	68.40
	STS Meth.	0.6	74.65	89.13	72.64
Lexicon-based	SyMSS (JCN)	0.45	74.70	84.17	70.87
	SyMSS (Vector)	0.45	74.15	90.32	70.82
	Omiotis	-	70.78	93.40	69.97
	LG (WUP)	0.6	73.90	91.07	71.02
Machine Learning-based	Wan et al.	-	77.00	90.00	75.00
	Z&P	-	74.30	88.20	71.90
	Qiu et al.	-	72.50	93.40	72.00
Baselines	Random	-	68.30	50.00	51.30
	VSM	0.5	71.60	79.50	65.40
Ours	PATH	0.6	73.86	91.37	72.75
	LCH	0.6	74.81	87.53	72.12
	WUP	0.6	71.07	94.25	70.67
	RES	0.6	74.79	87.18	71.94
	JCN	0.6	74.81	86.49	71.65
	LIN	0.6	74.81	86.49	71.65

5 Discussion

This section further discusses findings and issues of the STSS task and RTE, including the effects of POS sets and word measures.

5.1 Influence of POS Sets

Nouns and verbs can contribute main semantic features for short-text similarity. However, empirical evidence of this effect has not been provided thus far. In the training process, an experiment considered only nouns and verbs. The proposed method using PATH obtained the highest correlation values of 0.47 on

MSRpar, 0.442 on MSRvid, and 0.636 on SMTeur for the SemEval-2012 training datasets. For RTE, the proposed method using PATH achieved the highest accuracy value of 69.24% for the MSRpar training datasets.

In the performance test, similar results were found; nouns and verbs represented the main semantics for short-text similarity. For the SemEval-2012 test datasets, the proposed method using PATH obtained the highest correlations in each dataset: 0.465 on MSRpar, 0.733 on MSRvid, 0.516 on SMTeur, 0.656 on SMTnews, and 0.466 on OnWN. For RTE, the proposed method using PATH achieved the highest accuracy value 71.05% for the MSRpar test datasets. The results of both the training and test rounds suggest that nouns and verbs are the main features for short-text semantic analysis.

The proposed method obtained satisfactory accuracy for RTE (71.05%) by considering only nouns and verbs. However, it achieved similar performance to methods from previous works. The possible reason is that previous approaches [9, 12, 14] have used a single similarity measure to address all POSs. Because of the limitations of WordNet-based measures mentioned in Section 2.2, distance- and IC-based methods can evaluate noun and verb pairs, but they do not work for other POSs. The design of the proposed algorithm is consistent with these findings; the short-text similarity algorithm should categorize POSs and use corresponding measures for different POSs.

5.2 Influence of Word Measures

According to the conclusion of Section 5.1, using corresponding word measures address different POSs is a satisfactory approach for designing a short-text similarity algorithm. The idea is to use proper measures to address nouns and verbs because they contribute most semantics to a text. In the training process, this study designed the three POS sets and tested all the distance- and IC-based measures. In both training and performance testing, PATH achieved the best performance for nouns and verbs.

After comparing the results of first and second POS sets, the proposed algorithm achieved superior performance using VECTOR for adjectives and adverbs; thus, VECTOR is the correct measure for adjectives and adverbs. Furthermore, the proposed algorithm obtained the best performance by using the string measure for other POSs. Therefore, a short-text similarity algorithm should categorize POSs and use corresponding measures for different POSs.

Acknowledgements

This work was supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract No. MOST105-2511-S-415-017-MY3 and MOST105-2632-S-127-001.

References

[1] C. Chelmis, V. K. Prasanna, Predicting Communication Intention in Social Networks, *Proc. of the IEEE 2012 International Confernece on Social Computing*, Amsterdam, Netherlands, 2012, pp. 184-194.

[2] H.-H. Huang, H.-C. Yang, Semantic Clustering-based Community Detection in an Evolving Social Network, *Proc. of the IEEE 2012 Sixth International Conference on Genetic and Evolutionary Computing*, Kitakushu, Japan, 2012, pp. 91-94.

[3] Y. Xu, X. Guo, J. Hao, J. Ma, R. Y. K. Lau, W. Xu, Combining Social Network and Semantic Concept Analysis for Personalized Academic Researcher Recommendation, *Decision Support Systems*, Vol. 54, No. 1, pp. 564-573, December, 2012.

[4] J. Atkinson-Abutridy, C. Mellish, S. Aitken, Combing Information Extraction with Genetic Algorithms for Text Mining, *IEEE Intelligent Systems*, Vol. 19, No. 3, pp. 22-30, May-June, 2004.

[5] Y. Ko, J. Park, J. Seo, Improving Text Categorization Using the Importance of Sentences, *Information Processing and Management*, Vol. 40, No. 1, pp. 65-79, 2004.

[6] J.-W. Chang, M.-C. Lee, T.-I Wang, Integrating a Semantic-based Retrieval Agent into Case-based Reasoning Systems: A Case Study of an Online Bookstore, *Computers in Industry*, Vol. 78, pp. 29-42, May, 2016.

[7] Y. Liu, C. Zong, Example-based Chinese-English MT, *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, 2004, pp. 6093-6096.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-based Methods for Sentiment Analysis, *Computational Linguistics*, Vol. 37, No. 2, pp. 267-307, May, 2011.

[9] J. Oliva, J. I. Serrano, M. D. Castillo, Á. Iglesias, SyMSS: A Syntax-based Measure for Short-text Semantic Similarity, *Data & Knowledge Engineering*, Vol. 70, No. 4, pp. 390-405, April, 2011.

[10] G. A. Miller, WordNet: A Lexical Database for English, *Communications of the ACM*, Vol. 38, No. 11, pp. 39-41, November, 1995.

[11] A. Islam, D. Inkpen, Semantic Text Similarity Using Corpus-based word Similarity and String Similarity, *ACM Transactions on Knowledge Discovery from Data*, Vol. 2, No. 2, pp. 1-25, July, 2008.

[12] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, Text Relatedness Based on a Word Thesaurus, *Journal of Artificial Intelligence Research*, Vol. 37, No. 1, pp. 1-39, January, 2010.

[13] J. O'Shea, Z. Bandar, K. Crockett, D. McLean, A Comparative Study of Two Short Text, Semantic Similarity Measures, *Lecture Notes in Artificial Intelligence*, Vol. 4953, pp. 172-181, March, 2008.

[14] M.-C. Lee, J.-W. Chang, T.-C. Hsieh, A Grammar-based Semantic Similarity Algorithm for Natural Language Sentences, *The Scientific World Journal*, Vol. 2014, No. 437162, pp.1-17, April, 2014.

[15] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, K. Crockett, Sentence Similarity Based on Semantic Nets and Corpus Statistics, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 8, pp. 1138-1150, June, 2006.

[16] Z. S. Harris, *Mathematical Structures of Language*, Wiley, 1968.

[17] L. Han, T. Finin, P. McNamee, A. Joshi, Y. Yesha, Improving Word Similarity by Augmenting Pmi with Estimates of Word Polysemy, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 6, pp. 1307-1322, February, 2013.

[18] D. Klein, C. D. Manning, Accurate Unlexicalized Parsing, *Proc. of the 41st Annual Meeting on Association for Computational Linguistics*, Sapporo, Japan, 2003, pp. 423-430.

[19] D. D. Sleator, D. Temperley, *Parsing English with a Link Grammar*, CMU-CS-TR-91-126, August, 1995.

[20] R. Navigli, Word Sense Disambiguation: A Survey, *ACM Computing Surveys*, Vol. 41, No.2, pp. 1-69, February, 2009.

[21] S. Patwardhan, *Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness*, Ph. D. Thesis, University of Minnesota, Duluth, Minnesota, 2003.

[22] M. Lesk, Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pinecone from an Ice Cream Cone, *Proc. of the ACM 5th annual international conference on Systems documentation*, Toronto, Ontario, 1986, pp. 24-26.

[23] Z. Wu, M. Palmer, Verb Semantics and Lexical Selection, *Proc. of the 32nd annual meeting on Association for Computational Linguistics*, Las Cruces, New Mexico, 1994, pp. 133-138.

[24] C. Leacock, M. Chodorow, Combining Local Context and WordNet Similarity for Word Sense Identification, in: C. Fellbaum (Ed.), *WordNet: An electronic Lexical Database*, MIT Press, 1998, pp. 265-283.

[25] P. Resnik, Using Information Content to Evaluate Semantic Similarity in a Taxonomy, *Proc. of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Quebec, 1995, pp. 448-453.

[26] D. Lin, An Information-theoretic Definition of Similarity, *Proc. of the 15th International Conference on Machine Learning*, San Francisco, CA, 1998, pp. 296-304.

[27] J. J. Jiang, D. W. Conrath, Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, *Proc. of ROCLING X*, Taipei, Taiwan, 1997, pp. 19-33.

[28] I. Dagan, S. Marcus, S. Markovitch, Contextual Word Similarity and Estimation from Sparse Data, *Proc. of the 31st annual meeting on Association for Computational Linguistics*, Columbus, Ohio, 1993, pp. 164-171.

[29] E. Terra, C. L.A. Clarke, Frequency Estimates for Statistical Word Similarity Measures, *Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, Edmonton, Canada, 2003, pp. 165-172.

[30] N. Kaji, M. Kitsuregawa, Using Hidden Markov Random Fields to Combine Distributional and Pattern-based Word Clustering, *Proc. of the 22nd International Conference on Computational Linguistics-Volume 1*, Manchester, United Kingdom, 2008, pp. 401-408.

[31] T. Pedersen, S. Patwardhan, J. Michelizzi, WordNet: Similarity: Measuring the Relatedness of Concepts, *Proc. of HLT-NAACL 2004 (HLT-NAACL--Demonstrations'04)*, Boston, MA, 2004, pp. 38-41.

[32] M. P. Marcus, M. A. Marcinkiewicz, B. Santorini, Building a Large Annotated Corpus of English: The Penn Treebank, *Computational Linguistics*, Vol. 19, No. 2, pp. 313-330, June, 1993.

[33] E. Agirre, M. Diab, D. Cer, A. Gonzalez-Agirre, Semeval-2012 Task 6: A Pilot on Semantic Textual Similarity, *Proc. of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, Montréal, Canada, 2012, pp. 385-393

[34] H. Rubenstein, J. B. Goodenough, Contextual Correlates of Synonymy, *Communications of the ACM*, Vol. 8, No. 10, October, 1965, pp. 627-633.

[35] J. M. Sinclair, *Collins Cobuild English Dictionary for Advanced Learners*, Harper Collins, 2001.

[36] C. Quirk, C. Brockett, B. Dolan, Monolingual Machine Translation for Paraphrase Generation, *Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004, pp. 142-149.

[37] P. Turney, Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL, *Proc. of the Twelfth European Conference on Machine Learning*, Freiburg, Germany, 2001, pp. 491-502.

[38] R. Mihalcea, C. Corley, C. Strapparava, Corpus-based and Knowledge-based Measures of Text Semantic Similarity, *Proc. of the Twenty First Conference on Artificial Intelligence*, Boston, MA, 2006, pp. 775-780.

[39] S. Wan, M. Dras R. Dale, C. Paris, Using Dependency-based Features to Take the Parafarce out of Paraphrase, *Proc. of the Australasian Language Technology Workshop*, Sydney, Australia, 2006, pp. 131-138.

[40] Y. Zhang, J. Patrick, Paraphrase Identification by Text Canonicalization, *Proc. of the Australasian Language Technology Workshop*, Sydney, Australia, 2005, pp. 160-166.

[41] L. Qiu, M.-Y. Kan, T.-S. Chua, Paraphrase Recognition via Dissimilarity Significance Classification, *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006, pp. 18-26.

[42] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

Biographies

Po-Sen Huang received his B.S. in Marketing and a M.A. in Instructional Technology from San Jose State University, California respectively. He also received an Ed.S. in Curriculum & Instruction with emphasis in International Education at Kent State University, Ohio. His research interests include mobile learning, blended online collaboration, and multi-cultural studies.

Po-Sheng Chiu is currently an assistant professor of the Department of E-learning Design and Management, National Chiayi University, Taiwan. He received his Ph.D. degree in Department of Engineering Science from National Cheng Kung University, Taiwan, in 2013. His research interests include Educational Technology, Mobile Learning.

Jia-Wei Chang is an assistant professor in Department of Computer Science and Information Engineering, National Taichung University of Science and Technology. He received the Ph.D. degree from Department of Engineering Science, National Cheng Kung University in 2017. His research interests include internet of things, artificial intelligence, and e-learning technologies.

Yueh-Min Huang is a Chair Professor in the Department of Engineering Science, National Cheng-Kung University, Taiwan. His research interests include e-learning, wireless networks, artificial intelligence. He received his M.S. and Ph.D. degrees in Electrical Engineering from the University of Arizona in 1988 and 1991 respectively.

Ming-Che Lee received the Ph.D. degree from National Cheng Kung University, Taiwan, in 2008. Since 2008, he has been a professor at Computer and Communication Engineering of Ming Chuan University. His research interests are in the fields of Semantic Web, Ontology, Artificial Intelligence, Digital Content, Natural Language Processing.