Measuring Word Similarity Based on the Optimal Transformation and Combination of Features

ChukFong Ho, Masrah Azrifah Azmi-Murad, Shyamala Doraisamy, and Rabiah Abdul-Kadir

***Abstract*—The semantic similarity between two words can be determined based on their common and distinctive features after transforming them into measurable values. Up to now, a variety of transfer functions with respect to the transformation and the combination of features have been proposed. However, none of them have ever processed and combined those features properly, thus making them incapable of making a judgment of similarity that is close to human judgments of similarity. This paper offers a method that represents the optimal combination of the optimal transformation of two types of widely used features. The results obtained from a standard data set show that the proposed solution outperforms all of its benchmarks significantly.**

***Index Terms*—Semantic similarity, transfer function, word similarity.**

I. Introduction

WordNet is a lexical database that provides a deep insight regarding the meaning(s) of a word through the use of various semantic relations. These semantic relations link a word to another and form a hierarchical structure called Hierarchical Semantic Structure (HSS). Over the years, HSS has been used widely in the study of lexical semantics. A good example of this is the use of two features provided by HSS to determine the semantic similarity between words: the longest common subsumers (LCS) and the shortest path distance (SPD) where LCS represents the depth of their nearest common hypernyms and SPD represents the least number of paths connecting them. Having these features alone however, is not enough to make reliable judgments of similarity as Li et al. [11] remind us that the infinite properties possessed by them make direct use of them inappropriate.

The semantic similarity between words can be represented by a range of numerical values. Usually, these values ranged between 1 and 0, thereby creating the upper bound and lower bound constraints respectively. This means that two absolutely similar words should be assigned the value 1 while two absolutely dissimilar words should be assigned the value 0. However, when LCS and SPD are used directly, the upper bound and lower bound constraints could be violated owing to the fact that their values ranged between x and 0 where x is greater than 1. Put in other words, regardless of how similar or how dissimilar two words are, there are times they will be assigned a similarity score greater than 1. In order to deal with this issue, a transfer function is especially needed.

A transfer function is a function that transforms a feature into measurable values while satisfying both the upper bound and lower bound constraints. The effectiveness of a transfer function is dependent on the nature of its corresponding feature with respect to semantic similarity. For instance, to measure the semantic similarity between two words based on their LCS, a monotonically increasing transfer function should be used since their similarity is proportional to their LCS. However, there are too many possible monotonically increasing functions in mathematics. Without identifying the optimal transfer function among those possible transfer functions, LCS may not be transformed properly. As a result, an optimal judgment of similarity cannot be achieved.

Even if the optimal transfer function is used, solely relying on a single type of feature in making similarity judgment could be impractical because the contribution of features in measuring the semantic similarity between words may vary depending on many factors, for instance, how well a feature described the similarity relations and the quality of HSS. Moreover, Li et al. [11] found that when multiple features are used, a better judgment of similarity can be made. They also argue that when more than one feature is involved, the way of combining those features matters a lot. However, there are many possible types of combinations. Without identifying the optimal combination among the possible combinations, those features may not be combined properly. Consequently, an optimal judgment of semantic similarity cannot be made.

Therefore, it is possible to hypothesize that in order to produce judgments of similarity that are closer to human judgments of similarity, any method that relies on the use of multiple features must fulfill two optimality conditions: the optimal transformation and the optimal combination. It is worth noting that although a number of transfer functions and combinations have been suggested to measure the semantic similarity between words, no research has been found looking into these two optimality issues.

This paper is organized as follows: Section II highlights the importance of semantic similarity with respect to a wide range of natural language processing applications before presenting the overview of the related works. Section III and Section IV provide a detailed description of the optimal transformation and the optimal combination respectively and their significance is evaluated theoretically in Section V. Section VI describes the experimental setups and Section VII analyzes the experimental results followed by a discussion of the findings. Finally, Section VIII concludes this paper and summarizes the contributions of our work.

II. Related Works

Measuring the semantic similarity between words is

Manuscript received April 20, 2012, revised May 10, 2012.

Autors are with the Faculty of Computer Science and Information Technology, Selangor, Malaysia (e-mail: hochukfong@yahoo.com; masrah@fsktm.upm.edu.my; rabiah@fsktm.upm.edu.my; shyamala@fsktm.upm.edu.my).

usually an intermediate task rather than an end task by itself. This can be observed through its integration into a variety of natural language processing applications:

- In the detection of plagiarized documents [18], the similarity between two documents is calculated based on the similarity of their content words.
- In question answering [3], the answer for a question is identified by measuring the semantic similarity between question and answer.
- In the detection and correction of malapropisms [2], semantically incompatible words are considered as erroneous words.
- In text summarization [26] and machine translation [9], automatic evaluation is accomplished through the comparison of summaries based on sentence similarity.
- In text categorization [24], documents that share similar or related features are collected based on the semantic similarity between their content words.

In general, word similarity methods can be categorized into three classes: knowledge-based methods, corpus–based methods and hybrid-based methods. A knowledge-based method relies on human encoded knowledge of the semantic relations between words that are embedded in knowledge bases such as WorNet, HowNet and Roget's Thesaurus while a corpus-based method relies on syntax information collected from corpora while. A hybrid-based method relies on both the semantic information and syntax information.

A. Hybrid-Based Methods

Resnik [16] recommended a method that measures the information content (IC) of a word as the probability of its occurrence in a corpora where the difference between two IC represents the similarity between two words. Lin [12] also proposed an IC-based method. However, the IC is collected based on the occurrence of a word by taking its dependency relations into consideration. A major drawback of these two methods is that IC derived in this way is not applicable to polysemous words.

Jiang and Conrath [8] solved that problem by collecting IC from semantically disambiguated corpora. However, this kind of corpora is limited in availability.

B. Knowledge-Based Methods

Wu and Palmer [23] proposed a method based on LCS and SPD, however, their values are derived from the number of node instead of the number of path or link in HSS. Perhaps so far in the literature, this is the only node-based solution.

Sebti and Barfroush [19] modified the method suggested by Lin [12] by collecting IC from HSS rather than from corpora. Any calculated similarity scores that is deviated from human judgments of similarity are detected based on the combination of LCS and path distance and recalculated.

Yang and Powers [25] introduced a method based on SPD and the type of path. Although impressive results were reported, as a consequence of neglecting the importance of LCS, their method fails to differentiate any word pair with different LCS but with the same path distance and path type.

Li et al. [11] and Liu et al. [14] measured the semantic similarity between words based on LCS and SPD. However, none of their proposed methods satisfies the two optimality conditions as Li et al. [11] only focused on the transformation of LCS and SPD whereas Liu et al. [14] only focused on their combination.

Other than WordNet, several scholars have attempted to measure the semantic similarity between two words based on Roget's thesaurus [7] and HowNet [6]. However, WordNet has the benefit of wide applicability compared with Roget's thesaurus and HowNet. Besides, based on the results reported by Hu et al. [6], HowNet suffers from the coverage problem in terms of words.

In contrast to the above-mentioned methods that rely on the semantic relations between words, Liu et al. [13] and Wan and Angryk [22] suggested methods that measure the similarity of two words based on their definitions. After all, they measured the similarity between definitions as a function of the similarity between their content words, thus indicating the importance of using a word similarity method that is capable of producing reliable judgments of similarity.

C. Hybrid-Based Methods

Chen et al. [4] proposed a symmetric-based method based on web search. Given two semantically similar words, when one of them is used as a query, the other one can be found and vice versa. A major disadvantage of this method is that the author neglected the fact that a word may appear in different part of speech (POS) in different web documents.

Bollegala et al. [1] suggested the use of lexico-syntactic patterns and support vector machine (SVM) to improve the traditional web search-based method where synonymous and non-synonymous words provided by WordNet were used to train SVM. Despite the improvement, their proposed method still underperforms knowledge-based methods.

In conclusion, none of the previously proposed methods have ever taken the two optimality issues into consideration.

III. The Optimal Transformation

Among the previously proposed word similarity methods, only three methods [11, 14, 25] are managed to achieve Pearson's correlation coefficient (r) greater than 0.9 with respect to human ratings which is considered as high in this study. Hence, only the transfer functions and combinations belong to them are used to test the hypothesis.

A. Transfer Functions for SPD

$f(SPD)$ is a decreasing function that transforms SPD into a meaningful value in the range of 0 to 1 that tells how semantically similar two words are. Equation (1), (2) and (3) represent three transfer functions for SPD recommended by Yang and Powers [25], Li et al. [11] and Liu et al. [14] respectively.

$$f(SPD) = \begin{array}{ll} \alpha_t \prod_{i=1}^{l-1} \beta_{t_i} & l < \gamma \\ 0 & l \geq \gamma \end{array} \qquad (1)$$

$$f(SPD) = e^{-\lambda l} \qquad (2)$$

$$f(SPD) = e^{-\lambda l} - 1 \qquad (3)$$

For (1), given two words which can be located in HSS, l represents their SPD; t represents the type of path (synonym, hypernyms/hyponym or holonym/meronym); α_t represents the factor of path type; β_t represents the factor of path distance; γ represents an arbitrary threshold of the distance introduced for efficiency representing human cognitive limitations. The values of α_t, β_t and γ have already been tuned by Yang and Powers [25] where $\alpha_{synonym}$ equals to 0.9, both $\alpha_{hypernym/hyponym}$ and $\alpha_{holonym/meronym}$ equal to 0.85, β equals to 0.7, and γ equals to 12. For (2) and (3), λ represents the smoothing factors where their values have already been tuned as 0.25 by Li et al. [11] and Liu et al. [14] respectively.

According to the results published by Yang and Powers [25] and Li et al. [11], *r*(s) achieved using (1) and (2) equal to 0.921 and 0.891 respectively. If the hypothesis regarding the optimal transformation is correct, (1) should be a more optimal transformation for SPD than (2) since *r* achieved using (1) is greater than *r* achieved using (2). Also, when (1) and (2) are combined with any of the same transfer function for LCS separately, *r* achieved using (1) should always be greater than *r* achieved using (2). These two predictions derived from the hypothesis will be verified experimentally in Section VII.

B. Transfer Functions for LCS

f(*LCS*) is an increasing function that transforms LCS into a meaningful value in the range of 0 to 1 that tells how semantically similar two words are. Equation (4), (5) and (6) represent three transfer functions for LCS proposed by Li et al. [11] and Liu et al. [14] respectively:

$$f(LCS) = \frac{e^{\varepsilon d} - e^{-\varepsilon d}}{e^{\varepsilon d} + e^{-\varepsilon d}} \tag{4}$$

$$f(LCS) = e^{\varepsilon d} \tag{5}$$

$$f(LCS) = e^{\varepsilon d} - 1 \tag{6}$$

where *d* represents LCS of any two words which can be located in HSS and ε represents the smoothing factor. The value of ε in (4) has already been tuned as 0.15 by Li et al. [11] whereas the values of ε in (5) and (6) are tuned as 0.05 and 0.05 respectively using RG-65 (see Section VI).

IV. The Optimal Combination

The following combinations represented by (7), (8) and (9) were proposed by Liu et al. [14] and Li et al. [11]:

$$S(w_1, w_2) = \frac{f(LCS)}{f(LCS) + 1/f(SPD)} \tag{7}$$

$$S(w_1, w_2) = \delta f(LCS) + \vartheta f(SPD) \tag{8}$$

$$S(w_1, w_2) = f(LCS) \times f(SPD) \tag{9}$$

where δ and ϑ are the smoothing factors; w_1 and w_2 are words and $S(w_1, w_2)$ represents the semantic similarity between w_1 and w_2.

It is worth noting that (7) which was proposed by Liu et al. [14] contains two smoothing factors: λ (see (2) and (3)) and ε (see (5) and (6)). Other than smoothing the transformation of LCS and SPD (referring to *f*(*LCS*) and f(*SPD*) respectively), Li et al. also use λ and ε to correlate the combination of *f*(*LCS*) and *f*(*SPD*). Unless the values of the smoothing factors used for optimal transformation (or the internal smoothing factors) are equal to the values of the smoothing factors used for optimal combination (or the external smoothing factors), the two optimality conditions cannot be satisfied. For this reason, (10) which is a modified version of (7) and which consists of both the internal (λ and ε) and external (δ and ϑ) smoothing factors is introduced:

$$S(w_1, w_2) = \frac{\delta f(LCS)}{\delta f(LCS) + \vartheta / f(SPD)} \tag{10}$$

V. A Theoretical Investigation on Optimal Combination

As recalled in Section I, LCS and SPD between two words can be used to reflect their commonalities and differences respectively; and two absolutely dissimilar words should be assigned the value 0 whereas two absolutely similar words should be assigned the value 1. Taken together, these two statements are consistent with Liu et al.'s [14] constraints of semantic similarity:

- When there is no commonality between two words, they are absolutely dissimilar as *f*(*LCS*) = 0 when LCS = 0.
- When there is no difference between two words, they are absolutely similar as when *f*(*SPD*) = 1 when SPD = 0.

These constraints can be used to examine the significance of the three combinations provided in Section IV.

When LCS = 0, *f*(*LC*S) = 0. Then, when *f*(*LCS*) = 0, suppose $S(w_1, w_2) = 0$:

- Equation (8) = $\theta f(SPD)$. A study by Li et al. [11] reports that when LCS = 0, SPD tends to be a large value, that is, (8) ≈ 0.
- Equation (9) = 0.
- Equation (10) = 0.

When SPD = 0, *f*(*SPD*) ≈ 1. Then, when *f*(*SPD*) ≈ 1, suppose $S(w_1, w_2) \approx 1$:

- It is possible that (8) ≈ 1 if $\theta > \delta$. However, Li et al. [11] found that when (8) is used as the combination, $\delta > \theta$.
- It is possible to have (9) ≈ 1 if LCS is large enough that *f*(*LC*S) ≈ 1. However, based on the values of LCS and SPD reported in a study by Li et al. [11], when SPD = 0, LCS can be any value.
- It is possible that (10) ≈ 1 if the value of θ is small.

Based on the above analysis, it is apparent that only (10) satisfies both of the constraints of semantic similarity. Hence, it can be assumed that among the three tested combinations, (10) is the optimal combination for *f*(*SPD*) and *f*(*LCS*). Then, if our hypothesis regarding the optimal combination is true, when (10) is used to combine *f*(*SPD*) and *f*(*LCS*), *r* achieved using (10) should always be greater than *r*(s) achieved using (8) and (9). This prediction follows from the hypothesis will be investigated in Section VII.

VI. Experimental Design

A. Standard Data Set

The evaluation data set consists of 65 pairs of nouns and human ratings for their semantic similarity. This data set was created by Rubenstein and Goodenough [17] in 1965, and hence it came to be known as RG-65. Their experiment was then replicated by several scholars [15, 16] on 28 selected pairs of nouns, ended up creating another human rating called MC-28. Since after that, MC-28 and its corresponding RG-28 have widely been applied as the evaluation data set in the study of lexical semantics.

B. *Test Data Set and Training Data Set*

Due to the relatively small size of MC-28 and RG-28, one may question the validity of the obtained result. Therefore, an alternative machine learning technique called the repeated *k*-fold cross validation is applied on RG-65 and MC-28. In addition to that, our proposed method is also validated using 50 questions of ESL (English as a Second Language) test [20].

C. *Repeated k-Fold Cross Validation Technique*

To implement repeated *k*–fold cross validation, first, the test data set which consists of 30 definition pairs is randomly divided into *k* equally sized folds *j* times, thus creating *j* partitions of *k* folds. Then, for each partition of *j*, each fold is used as a test data set while the remaining *k*–1 folds are used as the training data set. For instance, when $fold_1$ is used as the test data set, $fold_2$ to $fold_k$ are used as the training data set, when $fold_2$ is used as the test data set, $fold_1$, $fold_3$ to $fold_k$ are used as the training data set, and so on. As a result, *k* correlations will be obtained and they are averaged. After these steps have been repeated on all of the *j* partitions, *j* averaged correlations will be obtained and they are averaged again. This averaged correlation represents the final correlation.

For RG–65, the value of *j* is set as equal to the value of *k* and {5, 13} are the possible values of *k*. A study by Kohavi [10] reports that the estimate of *k*–fold cross validation is good when k = 10. Therefore, among the two values, *k* = 13 is chosen since it is the nearest to *k* = 10. For MC–28, the value of *j* is set as equal to the value of *k* and {2, 4, 7, 14} are the possible values of *k*. However, *k* = 14 (and *k* = 2) is excluded due to the extremely small size of the resulted test data. Among the remaining two values, *k* = 7 is chosen since it is the nearest to *k* = 10.

D. *Experimental Conditions*

The semantic similarity between two words is measured under the following nine conditions:

- C1: *f*(*SPD*) = (1) and *f*(*LCS*) = (4).
- C2: *f*(*SPD*) = (1) and *f*(*LCS*) = (5).
- C3: *f*(*SPD*) = (1) and *f*(*LCS*) = (6).
- C4: *f*(*SPD*) = (2) and *f*(*LCS*) = (4).
- C5: *f*(*SPD*) = (2) and *f*(*LCS*) = (5).
- C6: *f*(*SPD*) = (2) and *f*(*LCS*) = (6).
- C7: *f*(*SPD*) = (3) and *f*(*LCS*) = (4).
- C8: *f*(*SPD*) = (3) and *f*(*LCS*) = (5).
- C9: *f*(*SPD*) = (3) and *f*(*LCS*) = (6).

Equation (9), (10) and (11) are evaluated by combining the transfer functions from these nine conditions.

VII. Results and Discussion

A. *The Optimal Transformation of LCS and SPD*

Table I presents different *r* achieved by different transfer function for LCS and SPD with respect to MC–28 judgments.

TABLE I: *R* Achieved by Different Transfer Function Individually

	f(*SPD*)			*f*(*LCS*)		
Equation	(1)	(2)	(3)	(4)	(5)	(6)
r	0.921	0.900	0.900	0.877	0.856	0.856

Since (1) and (4) achieve the highest *r* among the tested transfer functions for SPD and LCS respectively, (1) and (4) represent the optimal transformations for SPD and LCS respectively in the scope of this study. Since (1) achieves a higher *r* than (4), SPD could therefore be a major feature, if not the only one, describing the similarity relations between words.

However, back to 35 years ago, in year 1977, Tversky [21] claims that when measuring the semantic similarity between words, any feature that reflects their commonalities (i.e. LCS) plays a more important role than any feature that reflects their differences (i.e. SPD). 27 years later, Li et al. [11] demonstrated that LCS is more effective than SPD in producing similarity judgments.

Although the second finding is contrary to the earlier findings, it does not mean that either one must be wrong. Instead, the contrary findings could be suggesting that a more optimal transformation of LCS has not yet been discovered; thereby highlighting the significance of the hypothesis that when a non-optimal transfer function is used or when a feature is transformed improperly, the judgments of semantic similarity could not be optimized as well.

B. *The Optimal Combination*

Table II presents a set of *r*(s) achieved using different combinations: (8), (9) and (10). Since each of them combines the transfer functions for LCS and SPD under nine different conditions from C1 to C9, there are 27 combinations in total.

From Figure 1 and Figure 2, it is apparent that among the three combinations, only (8) and (10) achieve positive and yet consistent *r*(s) across the nine conditions. We also can see that among these conditions for (8) and (10), C1 achieves the highest *r*. For (10), although C7 achieves a higher *r* than C1 with respect to RG–28, the variance between them is small and therefore can be ignored.

However, further observation of Figure 1 and Figure 2 reveals that (10) achieves higher *r*(s) than (8) for all of the nine conditions. In addition to that, Table II also shows that (8) always gives more weight to SPD than LCS (i.e. $\delta > \vartheta$) which is contrary with the finding by Tversky [21]. Taken together, these findings suggest that (10) is a more optimal combination than (8).

As a summary, when (10) which represents the optimal combination is used to combine (1) and (4) which represent the optimal transformations for SPD and LCS respectively, the highest *r*(s) are achieved (see Table II, C1 of (10)). These findings are consistent with the outcome of the theoretical investigation in Section VI.

In conclusion, they verify the hypothesis that among all of the possible transfer functions, when the optimal transfer functions are combined using the optimal combination (or skeleton), an optimal judgment of similarity can be made. In the remaining part of this paper, C1 of (10) is referred to as word similarity method (WSM), representing our proposed solution: the optimal combination of the optimal transfer functions for LCS and SPD.

TABLE II: *R* VALUES ACHIEVED BY 27 COMBINATIONS OF THE TRANSFER FUNCTIONS OF LCS AND SPD

		C1	**C2**	C3	C4	C5	C6	C7	C8	C9
Equation 7.8		δ = 0.15 ϑ = 0.85	**δ = 0.1** **ϑ = 0.9**	δ = 0.1 ϑ = 0.9	δ = 0.15 ϑ = 0.85	δ = 0.05 ϑ = 0.95	δ = 0.05 ϑ = 0.95	δ = 0.15 ϑ = 0.85	δ = 0.05 ϑ = 0.95	δ = 0.05 ϑ = 0.95
r	RG-28	0.913	**0.911**	0.911	0.907	0.905	0.905	0.907	0.905	0.905
	MC-28	0.925	**0.923**	0.923	0.905	0.901	0.901	0.905	0.901	0.901
Equation 7.9		N/A	**N/A**	N/A	N/A	N/A	N/A	N/A	N/A	N/A
r	RG-28	0.868	**0.904**	0.828	0.861	0.896	0.817	N/A	0.888	N/A
	MC-28	0.890	**0.918**	0.851	0.872	0.896	0.827	N/A	0.879	N/A
Equation 7.10		δ = 0.85 ϑ = 0.25	**δ = 0.05** **ϑ = 0.9**	δ = 0.6 ϑ = 0.05	δ = 0.6 ϑ = 0.2	δ = 0.25 ϑ = 0.65	δ = 0.5 ϑ = 0.05	δ = 0.65 ϑ = 0.2	δ = 0.55 ϑ = 0.55	δ = 0.75 ϑ = 0.1
r	RG-28	0.914	**0.913**	0.912	0.909	0.907	0.905	0.916	0.909	0.914
	MC-28	0.931	**0.927**	0.929	0.918	0.904	0.917	0.924	0.907	0.926

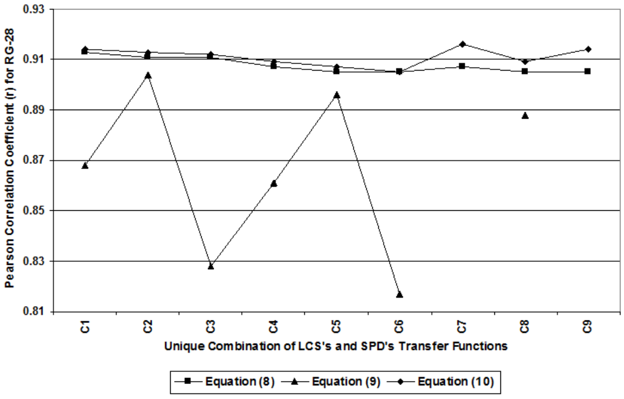


Fig. 1. *r* achieved by 27 combinations w.r.t. RG-28.

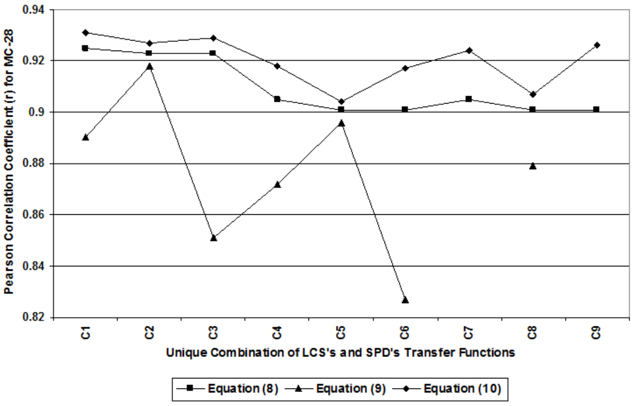
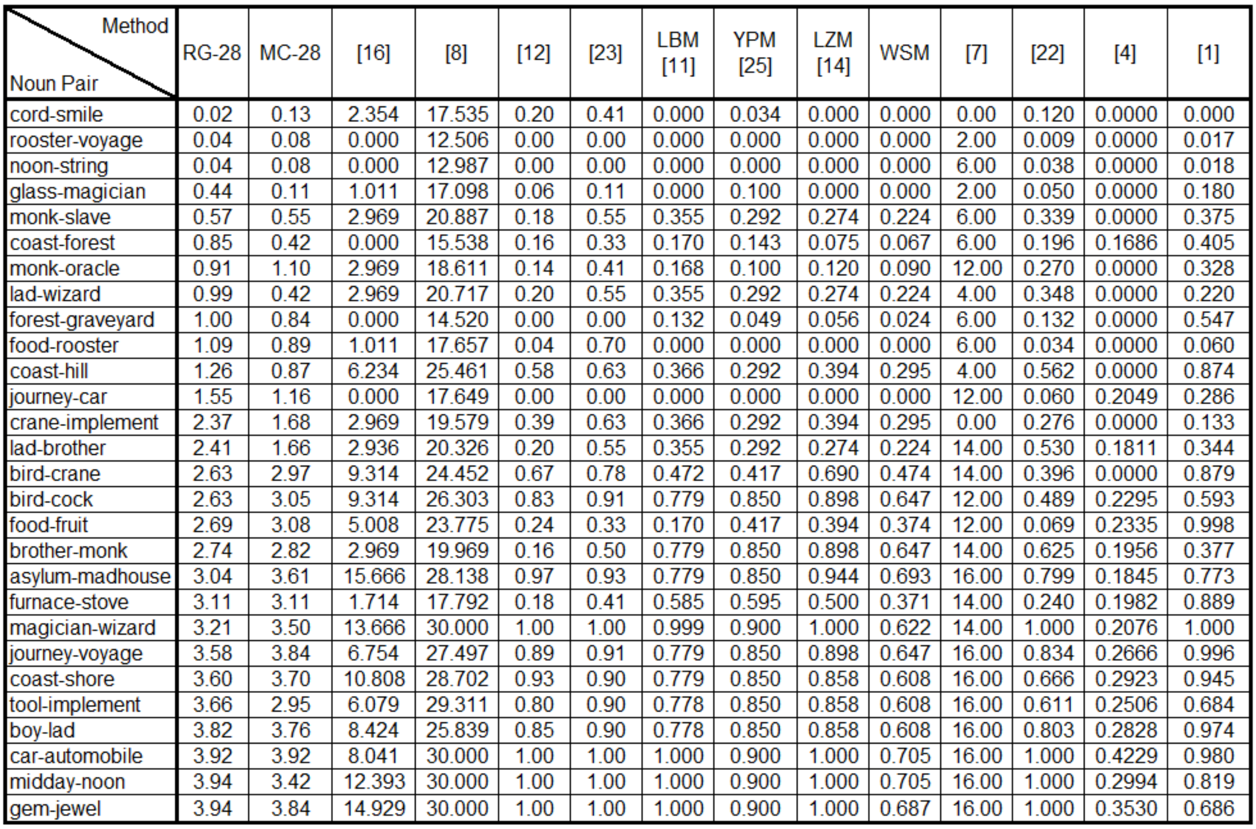


Fig. 2. *r* achieved by 27 combinations w.r.t. MC-28.

TABLE III: DATA SET RESULTS: SEMANTIC SIMILARITY SCORES



Method Noun Pair	RG-28	MC-28	[16]	[8]	[12]	[23]	LBM [11]	YPM [25]	LZM [14]	WSM	[7]	[22]	[4]	[1]
cord-smile	0.02	0.13	2.354	17.535	0.20	0.41	0.000	0.034	0.000	0.000	0.00	0.120	0.0000	0.000
rooster-voyage	0.04	0.08	0.000	12.506	0.00	0.00	0.000	0.000	0.000	0.000	2.00	0.009	0.0000	0.017
noon-string	0.04	0.08	0.000	12.987	0.00	0.00	0.000	0.000	0.000	0.000	6.00	0.038	0.0000	0.018
glass-magician	0.44	0.11	1.011	17.098	0.06	0.11	0.000	0.100	0.000	0.000	2.00	0.050	0.0000	0.180
monk-slave	0.57	0.55	2.969	20.887	0.18	0.55	0.355	0.292	0.274	0.224	6.00	0.339	0.0000	0.375
coast-forest	0.85	0.42	0.000	15.538	0.16	0.33	0.170	0.143	0.075	0.067	6.00	0.196	0.1686	0.405
monk-oracle	0.91	1.10	2.969	18.611	0.14	0.41	0.168	0.100	0.120	0.090	12.00	0.270	0.0000	0.328
lad-wizard	0.99	0.42	2.969	20.717	0.20	0.55	0.355	0.292	0.274	0.224	4.00	0.348	0.0000	0.220
forest-graveyard	1.00	0.84	0.000	14.520	0.00	0.00	0.132	0.049	0.056	0.024	6.00	0.132	0.0000	0.547
food-rooster	1.09	0.89	1.011	17.657	0.04	0.70	0.000	0.000	0.000	0.000	6.00	0.034	0.0000	0.060
coast-hill	1.26	0.87	6.234	25.461	0.58	0.63	0.366	0.292	0.394	0.295	4.00	0.562	0.0000	0.874
journey-car	1.55	1.16	0.000	17.649	0.00	0.00	0.000	0.000	0.000	0.000	12.00	0.060	0.2049	0.286
crane-implement	2.37	1.68	2.969	19.579	0.39	0.63	0.366	0.292	0.394	0.295	0.00	0.276	0.0000	0.133
lad-brother	2.41	1.66	2.936	20.326	0.20	0.55	0.355	0.292	0.274	0.224	14.00	0.530	0.1811	0.344
bird-crane	2.63	2.97	9.314	24.452	0.67	0.78	0.472	0.417	0.690	0.474	14.00	0.396	0.0000	0.879
bird-cock	2.63	3.05	9.314	26.303	0.83	0.91	0.779	0.850	0.898	0.647	12.00	0.489	0.2295	0.593
food-fruit	2.69	3.08	5.008	23.775	0.24	0.33	0.170	0.417	0.394	0.374	12.00	0.069	0.2335	0.998
brother-monk	2.74	2.82	2.969	19.969	0.16	0.50	0.779	0.850	0.898	0.647	14.00	0.625	0.1956	0.377
asylum-madhouse	3.04	3.61	15.666	28.138	0.97	0.93	0.779	0.850	0.944	0.693	16.00	0.799	0.1845	0.773
furnace-stove	3.11	3.11	1.714	17.792	0.18	0.41	0.585	0.595	0.500	0.371	14.00	0.240	0.1982	0.889
magician-wizard	3.21	3.50	13.666	30.000	1.00	1.00	0.999	0.900	1.000	0.622	14.00	1.000	0.2076	1.000
journey-voyage	3.58	3.84	6.754	27.497	0.89	0.91	0.779	0.850	0.898	0.647	16.00	0.834	0.2666	0.996
coast-shore	3.60	3.70	10.808	28.702	0.93	0.90	0.779	0.850	0.858	0.608	16.00	0.666	0.2923	0.945
tool-implement	3.66	2.95	6.079	29.311	0.80	0.90	0.778	0.850	0.858	0.608	16.00	0.611	0.2506	0.684
boy-lad	3.82	3.76	8.424	25.839	0.85	0.90	0.778	0.850	0.858	0.608	16.00	0.803	0.2828	0.974
car-automobile	3.92	3.92	8.041	30.000	1.00	1.00	1.000	0.900	1.000	0.705	16.00	1.000	0.4229	0.980
midday-noon	3.94	3.42	12.393	30.000	1.00	1.00	1.000	0.900	1.000	0.705	16.00	1.000	0.2994	0.819
gem-jewel	3.94	3.84	14.929	30.000	1.00	1.00	1.000	0.900	1.000	0.687	16.00	1.000	0.3530	0.686

C. Benchmark Overall Results

Table III provides the judgments made by different methods including WSM with respect to the similarity of 30 pairs of nouns as listed in the first column.

The second and the third columns present two different human judgments called RG-28 and MC-28 which have widely been used as the gold standard. Judgments made by WSM are contained in the 11th column while the remaining columns contain judgments made by previously proposed methods.

As recalled earlier, Edmonds and Hirst [5] argue for the presence of absolute and near synonyms and therefore absolute synonyms should be given the value of 1 representing maximum similarity but not near synonyms. However, as can be seen from Table III, for any two nouns that appear to be near synonyms of each other, most of the methods assign the value of 1 to them except for five methods: [1], [4], [16], YPM (the method proposed by Yang and Powers [25]) and WSM.

TABLE IV: PART OF THE DATA SET RESULTS CONTAINED IN TABLE III

Noun Pair	MC-28	YPM	WSM
bird-cock	3.05	0.850	0.647
brother-monk	2.82	0.850	0.647
asylum-madhouse	3.61	0.850	0.693
journey-voyage	3.84	0.850	0.647
coast-shore	3.70	0.850	0.608
tool-implement	2.95	0.850	0.608
boy-lad	3.76	0.850	0.608
magician-wizard	3.50	0.900	0.622
car-automobile	3.92	0.900	0.705
midday-noon	3.42	0.900	0.705
gem-jewel	3.84	0.900	0.687

Among those five methods, only WSM and YPM are managed to achieve high values of *r*. However, Table IV shows that provided the same set of nouns, the scores assigned by WSM are more numerous than those assigned by YPM, thus indicating that WSM can better differentiate the similarity between one pair of nouns from the others. It is likely therefore that compared with the judgments made by YPM, the judgments made by WSM are more similar to human judgments.

TABLE V: *R* ACHIEVED BY DIFFERENT METHODS W.R.T. RG-28 AND MC-28

Method	*r*	
	RG-28	MC-28
Hybrid-based Method		
[16]	0.743	0.791
[8]	0.842	0.836
[12]	0.822	0.834
Knowledge-based Method		
[23]	0.787	0.777
LBM [11]	0.912	0.912
YPM [25]	0.910	0.921
LZM [14]	0.909	0.926
WSM	0.914	0.931
[7]	0.851	0.872
[22]	0.815	0.793
Corpus-based Method		
[4]	0.852	0.828
[1]	0.797	0.834

Table V compares different *r* achieved by WSM and the other methods with respect to RG-28 and MC-28. Among these methods, WSM achieves the highest *r* of 0.914 and 0.931 with respect to RG-28 and MC-28. Since WSM adopts the skeleton of LZM (method proposed by Liu et al. [14]) to combine the transfer functions for SPD (of YPM [25]) and for LCS (of LBM, a method proposed by Li et al. [11]), WSM was benchmarked against these three methods. Results show that:

- For RG-28, WSM outperforms all of them at a significance level of 0.005.
- For MC-28, WSM outperforms LBM, YPM and LZM at a significance level of 0.05, 0.025 and 0.01 respectively.

TABLE VI: *R* ACHIEVED BY DIFFERENT METHODS W.R.T. RG-65 AND MC-28 USING REPEATED *K*-FOLD CROSS VALIDATION

Method	RG-65	MC-28
WSM	0.888	0.930
LBM	0.890	0.916
LZM	0.886	0.926

Table VI provides the results obtained by WSM, LBM and LZM using repeated *k*–fold cross validation on RG165 and MC-28. Notice that these results are almost identical with the one presented in Table V while Table VII shows that WSM, LBM and LZM achieve comparable results on ESL test.

TABLE VII: PERFORMANCE ON ESL TEST

Method	50 Questions	42 Questions
WSM	33/50 (66%)	33/42 (79%)
LBM	34/50 (68%)	34/42 (81%)
LZM	32/50 (64%)	32/42 (76%)

In conclusion, although the judgments made by LBM and LZM are also highly similar to human judgments, there are some drawbacks to using them in measuring the semantic similarity between words. In particular, LBM correlates the transfer functions for LCS and SPD using (9). However, it has already been verified theoretically and experimentally in Section V and Section VII respectively that (9) is not the proper way for combining the transfer functions for LCS and SPD. On the other hand, LZM correlates the combination of the transfer functions for LCS and SPD by giving more weight to SPD than LCS. However, doing so is not consistent with Tversky's [21] finding which showed that more weight should be assigned to LCS that reflects commonality in the assessment of similarity.

VIII. CONCLUSION

This paper has presented a method measures the semantic similarity between words by taking takes into consideration two optimality conditions: the optimal transformation and the optimal combination. The experimental results show that the proposed method outperforms its benchmarks statistically significant at 0.05 levels, thereby underlining the importance of satisfying the two optimality conditions. In spite of the improvement, the optimal transformation and combination are only limited to two types of widely used features. In fact, there are many more features which have already been explored by the other scholars. As a future work, it is possible to extend the proposed optimal combination so that it can accommodate any number and any type of features.

ACKNOWLEDGMENT

We thank A. Jaafar for giving some mathematical advice on different combinations in terms of mathematic equations.

REFERENCES

[1] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring Semantic Similarity between Words Using Web Search Engines", in *Proc. 16th international conference on World Wide Web*, 2007, pp. 757–766.

[2] I. A. Bolshakov, "An Experiment in Detection and Correction of Malapropisms Through the Web", *Lecture Notes in Computer Science*, vol. 3406/2005, pp. 803-815, 2005.

[3] M. D. Boni, and S. Manandhar, "An Analysis of Clarification Dialogue for Question Answering", in *Proc. HLT-NAACL*, 2003, pp. 48-55.

[4] H. H. Chen, M. S. Lin, and Y. C. Wei, "Novel Association Measures Using Web Search with Double Checking", in *Proc. 21st International Conference on Computational Linguistics and 44th Annual Meetings of the ACL*, 2006, pp. 1009–1016.

[5] P. Edmonds, and G. Hirst, "Near–Synonymy and Lexical Choice", *Computational Linguistics*, vol. 28, no. 2, pp. 105–144, 2002.

[6] J. W. Hu, L. L. Dai, and B. Liu, "Measure Semantic Similarity between English Words", in *Proc. 9th International Conference for Young Computer Scientists*, 2008, pp. 1689–1694.

[7] M. Jarmasz, and S. Szpakowicz, "Roget's Thesaurus and Semantic Similarity", in *Proc. Conference on Recent Advances in Natural Language Processing*, 2003, pp. 212–219.

[8] J. J. Jiang, and D. W. Conrath, "Semantic Similarity Based on Corpus Statistic and Lexical Taxonomy", in *Proc. International Conference Research on Computational Linguistics* (*ROCLING X*), 1997.

[9] D. Kauchak, and R. Barzilay, "Paraphrasing for Automatic Evaluation", in *Proc. Human Language Technology Conference of the North American Chapter of the ACL*, 2006, pp. 455-462.

[10] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", in *Proc. 14th International Joint Conference on Artificial Intelligence* (*IJCAI*), 1995, pp. 1137-1143.

[11] Y. H. Li, Z. A. Bandar, and D. McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources", *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 871–882, 2003.

[12] D. Lin, "An Information-Theoretic Definition of Similarity", in *Proc. 15th International Conference on Machine Learning*, 1998, pp. 296–304.

[13] H. Liu, J. Zhao, and R. Lu, "Computing Semantic Similarities based on Machine–Readable Dictionaries", in *Proc. IEEE International Workshop on Semantic Computing and Systems*, 2008, pp. 8–14.

[14] X. Y. Liu, Y. M. Zhou, and R. S. Zheng, "Measuring Semantic Similarity in WordNet", in *Proc. Sixth International Conference on Machine Learning and Cybernetics*, 2007, pp. 3431–3435.

[15] G. A. Miller, and W. G. Charles, "Contextual Correlates of Semantic Similarity", *Language and Cognitive Processes*, pp. 1–28, 1991.

[16] P. 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, pp. 95–130, 1999.

[17] H. Rubenstein, and J. B. Goodenough, "Contextual Correlates of Synonymy", *Communications of the ACM*, pp. 627–633, 1965.

[18] S. Sandhya, and S. Chitrakala, "Plagiarism Detection of Paraphrases in Text Documents with Document Retrieval", *Communications in Computer and Information Science: Advances in Computing and Information Technology*, pp. 330–338, 2011.

[19] A. Sebti, and A. A. Barfroush, "A New Word Sense Similarity Measure in WordNet", in *Proc. International Multiconference on Computer Science and Information Technology*, 2008, pp. 369–373.

[20] D. Tatsuki, "Basic 2000 Words – Synonym Match 1", *English Vocabulary Quizzes – Medium* (*for ESL, EFL students*), 1998.

[21] A. Tversky, "Feature of Similarity", *Psychological Review*, vol. 84, pp. 327–352, 1977.

[22] S. Wan, and R. A. Angryk, "Measuring Semantic Similarity Using WordNet-based Context Vectors", in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 908–913.

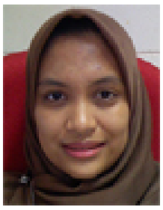
[23] Z. B. Wu, and M. Palmer, "Verb Semantics and Lexical Selection", in *Proc. 32th Annual Meeting of the Association for Computational Linguistics*, 1994, pp. 133–138.

[24] C. Yang, and J. Wen, "Text Categorization Based on a Similarity Approach", in *Proc. International Conference on Intelligent Systems and Knowledge Engineering* (*ISKE*), 2007.

[25] D. Q. Yang, and D. M. Powers, "Measuring Semantic Similarity in the Taxonomy of WordNet", in *Proc. 28th Australasian Computer Science Conference*, 2005, pp. 315–332.

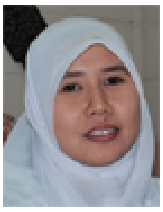
[26] L. Zhou, and C. Y. Lin, D. S. Munteanu, and E. Hovy, "ParaEval: Using Paraphrases to Evaluate Summaries Automatically", in *Proc. Human Language Technology Conference of the North American Chapter of the ACL*, 2006, pp. 447-454.

ChukFong Ho obtained his Bachelor of Computer Science from University Putra Malaysia in 2007. Currently, he is doing his PhD at the Faculty of Science Computer and Information Technology, University Putra Malaysia. His field of study is natural language processing and his interests are paraphrase extraction, paraphrasing, semantic word similarity and semantic sentence similarity.



Masrah Azrifah Azmi-Murad is an Associate Professor at the Department of Information System, University Putra Malaysia (UPM). She received her Bachelor of Management Information Systems from Drexel University, Philadelphia, USA in 1997 and her Master of Computer Science from University Kebangsaan Malaysia (UKM) in 1999. In 2000, she joined UPM as a tutor before granted a study leave to pursue the degree of PhD in the Department of Engineering Mathematics, University of Bristol, UK from 2001 to 2005. Then, she became a lecturer at the Department of Information Systems, UPM until present. Her areas of specialization are information retrieval, artificial intelligence and text mining.

Shyamala Doraisamy is an Associate Professor at the Department of Multimedia, UPM. She received her Bachelor of Computer Science from University of Science Malaysia in 1988 and her master of Information Technology from University Malaysia Sarawak (UNIMAS) in 1995. She joined the Department of Multimedia, UPM in 1998 as a lecturer. From 2000 to 2004, study leave was granted to undertake a postgraduate research program towards the degree of PhD at the Department of Computing, Imperial College London. Her areas of specialization are information retrieval (text, music and image) and audio mining.



Rabiah Abdul-Kadir is a Senior lecturer at the Department of Multimedia, UPM. She received her Bachelor of Computer Science from UPM in 1993 and her master and PhD of Computer Science from UKM. During the study of her Master, she worked as a tutor in UPM. After that, she joined the Department of Multimedia, UPM as a lecturer in 1997 until present. Her areas of specialization are computational linguistic, natural language understanding, artificial intelligent system and information access.