Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 3779 – 3790

doi:10.3906/elk-1901-91

Sentence similarity using weighted path and similarity matrices

Reza JAVADZADEH, Morteza ZAHEDI*, Marzea RAHIMI
School of Computer and IT Engineering Shahrood University of Technology Shahrood, Iran

Received: 12.01.2019 • **Accepted/Published Online:** 18.07.2019 • **Final Version:** 18.09.2019

Abstract: Sentence similarity is the task of assessing how similar the two snippets of text are. Similarity techniques are used extensively in clustering, summarization, classification, plagiarism detection etc. Due to a small set of vocabularies, sentence similarity is considered to be a difficult problem in natural language processing. There are two issues in solving this problem: (1) Which similarity techniques to be used for word pair similarity and (2) How to generalize that to sentence pairs. We have used the weighted path, a WordNet-based similarity assessment, and the paraphrase database to obtain word pair similarity values. Thereafter, we extracted maximum values from the pairwise similarity matrix and computed a similarity value for a sentence pair. We have also incorporated a vector space model technique to form a robust similarity measure. Our method outperformed state-of-the-art methods on the STSS65 test dataset in Pearson's correlation of 87% compared to human similarity scores. Moreover, our approach performed on par with other methods on the STSS131 test data using the same test. Our approach outperforms all the other WordNet-based methods compared on both datasets.

Key words: Sentence similarity, plagiarism detection, text mining, vector space model, paraphrase database

1. Introduction

Similar sentences may discuss the same idea, or they may be on a similar topic. Similar sentence pairs usually contain common words, link to common concepts, and have many cooccurring words. In sentence similarity assessment, given a pair of sentences, the task is to determine a similarity score between zero (the sentences are not equal in meaning) and four (the sentences are identical in meaning). The scores are then normalized between zero and one. The goal is to design a system that can assign scores to pairs such that they would have high correlation with human similarity scores. There are many applications for these methods, including clustering [1], summarization [2], classification [3], and word sense disambiguation [4]. We have divided the previous findings into multiple categories: statistical approaches, WordNet word-pair similarity methods based on edge-path distance, the systems based on dependency parsers, alignment solution methods, and vector space models (VSM) systems. However, there were systems that did not belong to a specific category; thus, we merged them within a similar category, and also note that there are overlaps between the methods.

BLEU metric is commonly used to evaluate machine translation systems. It considers the number of n-gram overlaps to assess a model's translation [5]. However, this system has a naive assumption to assess as it only considers the identical word overlaps [6]. Latent semantic analysis (LSA) is a popular approach which is used extensively for NLP tasks [7]. This method is based on statistics and uses the frequency values of words in both sentences to compute similarity. This approach decomposes the original cooccurrence matrix of

*Correspondence: Zahedi@shahroodut.ac.ir

words per documents into the singular value decomposition matrices. In other words, this way we can move the axis of space in the direction of highest variance in which most of the data points are scattered. This way we can segment words into meaningful clusters for further comparison. However, LSA is a linear model and it cannot model nonlinear relationships and the results are difficult to interpret. Second order cooccurrence PMI (SOC-PMI) uses the frequency information of common context words around each word pair to compute word similarity [8]. Semantic text similarity (STS) [9] proposes to combine three metrics to compute similarity: (1) string matching in which the number of common characters between word pairs is computed, (2) the SOC-PMI approach, and (3) word order information. This method is language-independent; however, it has data sparsity problem and a similar word pair does not necessarily share common characters. STS is based only on statistical information. Cho's method [10] has used the cooccurrences of words between the pairs to compute similarity. Although this method is language-independent, it has data sparsity problem and as such it is very difficult to compute a precise similarity value solely on context words. Our approach differs from STS in that it uses the weighted path (WP) (a WordNet-based metric) as a function of similarity. This metric considers both hierarchical information and statistical information; however, in STS only sparse statistical information was computed. Also, we incorporate our approach with a vector space model to produce better output against human scores. On the other hand, we used the paraphrase database to compute word pair similarities for the VSM model, which provides an improvement over the WordNet-based measure.

WordNet, in [11], introduces a semantic thesaurus hierarchy in which words are categorized into concepts. For example "male" and "female" are categorized into "human". In 2014, the researchers argued that WordNet-based metrics do not consider the importance of the shared concept's depth's to the root in the hierarchy. This meant that other metrics only considered the depth of the least common subsumer or the shortest path length, which was not sufficient for a precise similarity judgment. Ghazizadeh et al. [12] improved word pair similarity by giving weights to the edges of WordNet hierarchy, giving higher weights to the concepts away from root. However, they used WordNet 2.1 but we use YagoNet which is an extended version of WordNet and contains more information such as more name entities and Wikipedia. Also, our metric considers the information content value of the common concept which proved to be crucial by research in 2018 [6]. Omiotis [13] is another approach based on WordNet, which uses extra information such as synonymy and polysemy of words to compute similarity. Furthermore, part of speech (POS) tag information is used to weight the contribution of each part of speech. Although this method produced good results, it is not always proper to assign a constant weight for a certain POS tag as the contribution of each may differ in other sentence pairs. In a study in 1999, researchers used the information content of the common concept between word pairs to compute similarity [14]. However, it missed out on other important information such as the depth and the shortest path length between two concepts. In concept transferred space (CTS), all word types are mapped to nouns and then to concepts [6]. The information content of the common concept between word pairs are used to compute similarity and name entities are recognized using Stanford CoreNLP tools [15]. However, this method does not consider the path length between concepts and treats all concepts equally, which makes it less efficient. In fact, the idea of using a common concept to compute similarity between multiple words goes back to 2016 [16]. However, in that study, name entity recognition was not considered. Modifications were made to determine the value of the common concept in 2017 [17]. Furthermore, they enhanced their approach to CTS in 2018.

The SymSS method produces a dependency parser for each sentence, then words on the corresponding leafs are compared for similarity. A penalty factor was also defined to penalize the words which do not have corresponding leaves on the other sentence. Then the results are averaged to produce a final similarity value

[18]. The disadvantage of this method is that no optimal alignment was defined to compute similarity and this method does not consider all possible word pairs, which leads to missing important information. Minkov and Cohen [19] computed word similarity by considering dependency-parsed text as an instance of a labeled directed graph wherein words are the nodes and edges represent word relations. Iosif and Potamianos [20] created a corpus of web documents and extracted statistical word information to compute similarity.

Sultan's approach is an unsupervised alignment of two sentences. The ratio of aligned words is considered to be the ratio of similarity [21]. Dynamic time warping is another technique with a similar goal which was originally used to find patterns in time series [22]. However, such alignment techniques do not produce optimum results when there are no common words between a sentence pair. In 2007, the researchers used dynamic time warping (DTW) technique to compute how much change is needed to convert one sentence to another [23]. DTW is a distance metric, which allows similar signals to match. They defined a WordNet-based measure to compute the distance between word pairs. However, a given sentence can differ in many ways other than a simple word position change, this approach does not provide a robust measure for similarity assessment. The Needleman–Wunsch algorithm, introduced in [24], was first used for finding similarities in the amino acid sequences of two proteins. However, it can also be used for aligning two snippets of texts. That is, using this method we can measure how many actions, including substitutions, deletion, insertion it would take to convert one sentence to another. Feng's [25] approach first extends words to their synsets, then computes the similarity value between the words with the same part of speech and adds the result to compute direct similarity. It also computes the Needleman–Wunsch distance as a metric of indirect relevance. The two measures were then combined to form a single similarity measure. However, the Needleman–Wunsch distance cannot make a correct comparison as the sentence pair may not share many common words.

Lightweight semantic similarity (LSS) [26] is another approach that creates a list containing distinct sets of words. Hence, a feature vector is formed for each sentence by computing the similarity value between each word in the sentence and in the list. STASIS [27] attaches two sentences to form a list containing n distinct sets of words. Thereafter, for each word in the list, a word with maximum similarity is found in sentence A. This leads to an n-dimensional feature vector for sentence A containing similarity values. The same procedure is done for sentence B. The final similarity value is computed using the cosine measure. Similarity of word order between the pair is also considered to be important in the STASIS method. This method also has data sparsity problem.

Most researchers investigating similarity have utilized only one approach for feature extraction from similarity matrices. None of the recent approaches have used weighted path metric to compute similarity between sentences. Thus the contributions of the current study can be summarized as follows:

- Two methods, namely sentence vector similarity and similarity matrix values, were combined to form a robust measure. This has led to a better correlation compared to their individual results.
- A YagoNet-based metric is used to compute similarity which considers both the information content and the shortest path length between two concepts.
- The usage of YagoNet ensures a higher coverage of name entities for similarity assessment in comparison to other methods.
- Paraphrase database (PPDB) information were mixed with weighted path metric to cover an extensive number of word pair similarities. The addition of PPDB information ensures that both statistical measures and hierarchical-based similarity were considered in similarity assessment.

The rest of the paper is organized as follows: In Section 2, we have described our proposed approach. In Section 3, we compare our approach with the state-of-the-art approaches and finally in Section 4, we conclude our work and suggest a future work.

2. The proposed approach

In this section, we describe two methods to tackle the problem of sentence similarity assessment. Our approach comprises two main methods: (1) weighted path metric and extracting maximum values from the similarity matrix and (2) paraphrase database and cosine similarity between sentence feature vectors. We use Yago thesaurus as the source of information. Yago is an extension of WordNet and it provides more information using Wikipedia and GeoNames [1] [28]. At the preprocessing step, stop words were removed and words were lemmatized using Stanford CoreNLP tools.

2.1. Weighted path

This metric, introduced by Zhu and Iglesias, is computed on YagoNet using Equation 1 [29]:

$$S_{wpath}(C_1, C_2) = \frac{1}{1 + length(C_1, C_2) \times k^{IC(C_{lcs})}}. \tag{1}$$

Here *length* is the shortest path connecting C_1 and C_2 and $IC(C_{lcs})$ is the information content value of the ancestor of both C_1 and C_2. Information content value (IC) is based on the frequency of the common concept which is precomputed on the British National Corpus. Thus, there is no computation overhead for this measure. k controls the contribution of information content. This value should be in the range of [0,1]. The optimum value is found to be 0.8 during trial and error on the training data. IC is computed using Equation 2:

$$IC(C) = -log(P(C)) = \frac{\sum_{w_i \in C} count(w)}{N}. \tag{2}$$

Here $P(C)$ is the occurrence probability of concept C. $\sum_{w_i \in C}$ is the summation over all words which are subsumed by concept C and N is the total number of concepts in the corpus. After computing similarity for all word pairs we obtain a matrix containing all values as in Equation 3. Similarity for identical word pairs were not computed as they were put aside for final similarity computation. Thus, δ in Equation 3 is the number of identical words in the sentence pair:

$$M = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{(m-\delta)1} & \alpha_{(m-\delta)2} & \cdots & \alpha_{(m-\delta)(n-\delta)} \end{bmatrix}. \tag{3}$$

Here m and n are the number of words in the first and second sentence respectively. For example, α_{13} denotes the similarity value between the first word of the first sentence and the third word of the second sentence. Next step is feature extraction from this matrix. From this matrix maximum similarity values were extracted according to the following procedure:

1. The maximum similarity value is found and added to a list called p.

[1] GeoNames (2018). GeoNames geographical database. Website: https://www.geonames.org. [accessed: 25 July 2019]

Table 1. The procedure to compute a feature vector for each sentence pair.

	Automobile	Car	Motor	Vehicle	Room	Small	Number	Passenger
Automobile	1	0	$< 0 >$	0	0	0	0	0
Car	0	1	$< 0.73 >$	0	0	0	0	0
Car	0	1	0.73	$<< 0 >>$	0	0	0	0
Motor	0	0.73	1	$<< 0.73 >>$	0	0	0	0
Vehicle	0	0	0.73	$<< 1 >>$	0	0	0	0
Room	0	0	0	$<< 0 >>$	1	0	0	0
Small	0	0	0	$<< 0 >>$	0	1	0	0
Number	0	0	0	$<< 0 >>$	0	0	1	0
Passenger	0	0	0	$<< 0 >>$	0	0	0	1
Sentence A	1	1	$< 0.73 >$	0	0	0	0	0.0
Sentence B	0	1.73	2.46	$<< 1.73 >>$	1	1	1	1

2. The corresponding row and column are removed.

Thus, the procedure should be repeated until no more element in the matrix remains. The final similarity value is computed according to Equation 4, where m and n are as described earlier.

$$S_1 = \frac{\delta \sum_{i=1}^{|p|} p_i \times (m+n)}{2mn} \tag{4}$$

The usage of the Hungarian method [30] did not result in better correlation. Moreover, our approach may further enhance the feature extraction process by selecting more than one maximum value from each row for a future work.

2.2. Paraphrase database

The paraphrase database includes millions of synonym word pairs. We have used PPDB-2.0-xxxl version single-word-to-single-word synonyms [31, 32]. This database is based on a typical five-level Likert system which means the scores are as the following:

1. Strongly disagree (that the two words are equivalent in meaning)
2. Disagree
3. Neither agree nor disagree
4. Agree
5. Strongly agree

Firstly, all words in both sentences are added to a list. For example, in the case of *an automobile is a car* and *a car is a motor vehicle with room for a small number of passengers*, Table 1 is formed after removing the stop words:

As is shown in Table 1, the values inside "<>" in sentence A were added to form the corresponding feature value for sentence A and similarly the values inside "<<>>" were added to form a feature vector

for sentence B. The similarity value for this part is computed using Equation 5. The final similarity value incorporating the two methods is the average value obtained from both Equation 4 and Equation 5.

$$S_2 = cos(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{5}$$

3. Experimental results

Yago similarities were computed using sematch toolbox in Python 2.7. All WordNet similarities were computed using NLTK toolbox in Python 3.7. STASIS code was written using Python 3.7. All the other codes were written using Java 1.8. Two datasets were used to evaluate and compare our proposed approach with recent methods. Note that for some methods in experiments we could not provide comparison values as they were unavailable to our usage (either the code or the results).

3.1. Dataset

The STSS65 dataset includes 35 training sentence pairs and 30 test pairs [33]. The subjects of pairs vary in general topic or a description of facts. For example, *A hill is an area of land that is higher than the land that surrounds it* and *Woodland is land with a lot of trees.* As in the example, a sentence length may vary from a few to many sets of words. The STSS131 dataset includes 36 sets of training samples and 30 test pairs [34]. The subject of pairs seems to be a generic topic of conversations. For example, *If you don't console a friend, there is a chance you may hurt their feelings* and *One of the qualities of a good friend is the ability to console.* The sentence length is on average greater than that of STSS65. In the following sections, we compare our work with other recent researches.

3.2. Correlation results

In Figure 1, we compare our proposed approach (weighted path) with the state-of-the-art approaches both in terms of the Pearson's and Spearman's correlation coefficients on the STSS65 dataset. In Figure 2, we show the comparison results on the STSS131. We could not provide the Spearman's correlation value for CTS as it was not reported in the corresponding literature [6].

The results demonstrate that our proposed approach outperforms other methods according to Pearson's correlation against human scores. Therefore, our approach has a strong linear relationship with target values. The CTS method [6] stands next while Omiotis has performed best according to Spearman's correlation. In this problem, the Pearson's metric is considered to be more important [35]. In our approach and CTS, the information content value of the ancestor concept is considered as a measure of similarity and all methods (except for STS) have used WordNet-based metric. This seems to be the current superior approach; however, using such measures mean that the solution is language-dependent. The STS approach uses three metrics and all metrics are language-independent. Feng's approach has obtained comparable results and using the Needleman–Wunsch method seems to be the most significant difference between Feng's work and those of the others. On the STSS131 test dataset, the results are shown in Figure 2.

On the STSS131 dataset, there could be a number of common words between the pairs yet they may not bear the same meaning; this may have been the reason of decreased correlation in all approaches. LSA has performed best while our method stands next to LSA on this dataset. LSS performed best on Spearman's test which means that it is superior in terms of statistical dependence between the rankings of target and output

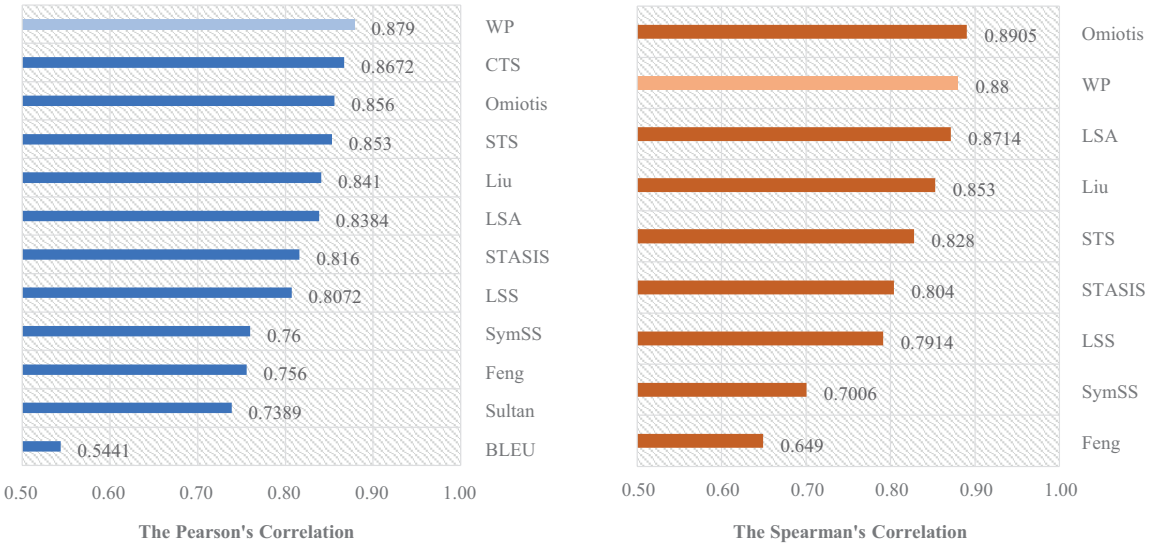

Figure 1. The comparison of our approach on the STSS65 test data.

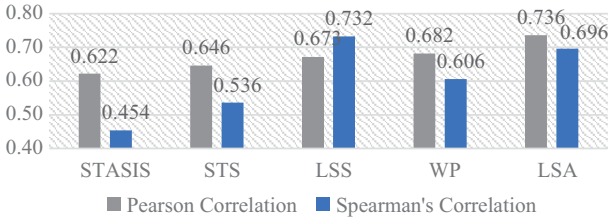

Figure 2. The comparison of our approach on the STSS131 test data.

variables. STASIS performed poorly on this dataset. This could be due to depending solely on a WordNet-based measure to form a relative sentence vector. This feature vector may not contain all information about a sentence to do a precise similarity assessment.

3.3. Mean deviation

We have compared our approach to recent ones in terms of the mean deviation of the automatically assigned scores from human similarity scores. This metric is computed using Equation 6 and the results are given in Figure 3. The best approach is the one with the smallest value.

$$md = \frac{1}{n}\sum_{i=1}^{n}|X_i - Y_i|. \tag{6}$$

Our approach performs better than the state-of-the-art methods on this test. The STS method performs on par with our approach while Feng's method performed surprisingly well on this test. The LSA method has obtained the highest value on this test which means that the assigned values were far off from the target scores in average.

Results on the STSS131 dataset demonstrates that our work, LSA, and STS are on par with human similarity scores, which means that these three approaches have the least average difference from human scores. STASIS performed poorly while LSS performed significantly worse on this test, which makes the good results it obtained in terms of correlation questionable. Therefore, we further test this approach to determine its performance.

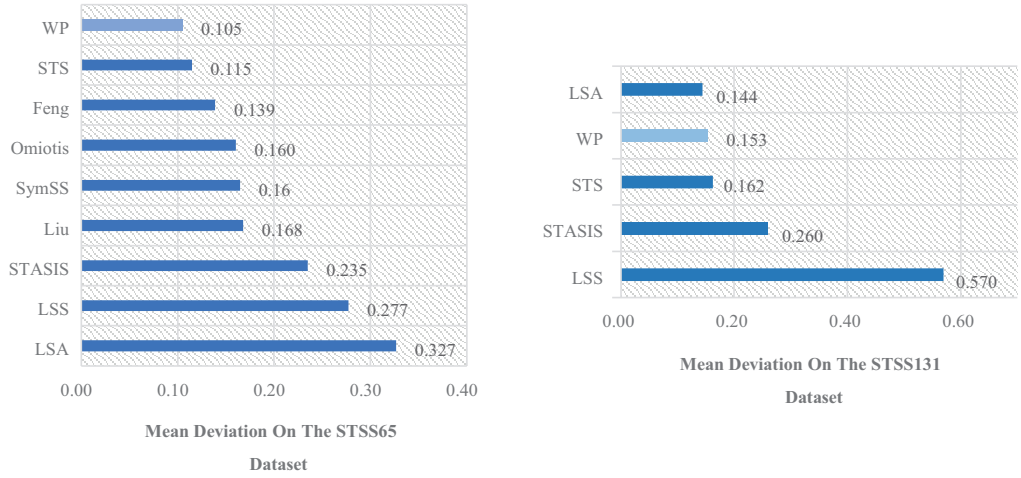

Figure 3. The comparison of our approach in terms of mean deviation on the STSS65.

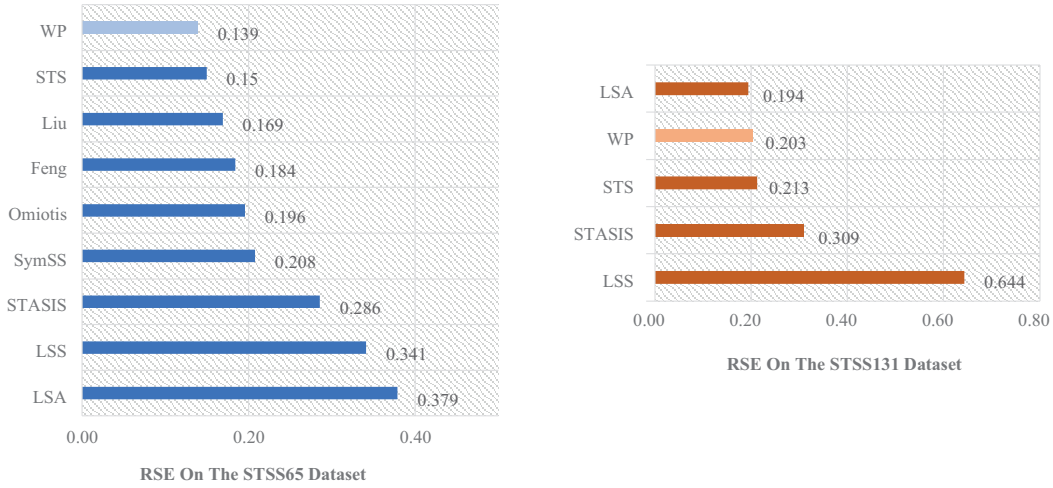

Figure 4. The comparison of our approach with recent methods in terms of RSE

3.4. Residual standard error

Due to possible noises in the procedure of feature extraction and lack of enough number of features, the target regression line may not be predicted with perfect precision. Residual standard error (RSE) can measure the amount of imperfection of a prediction model, so the goal is to design a model with the least possible RSE. Figure 4 demonstrates the comparison of our approach with recent ones on the STSS65 dataset. This measure is computed using Equation 7 [36].

$$RSE = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{7}$$

The best feature sets seem to be considered in our proposed approach as we have computed two types of features from pairwise similarity matrices namely maximum values and summation. Our approach has obtained yet again the best results on this test while STS follows similar results. LSA has obtained the highest value on this test while Omiots, Feng, and STASIS are three other WordNet-based methods which have obtained similar

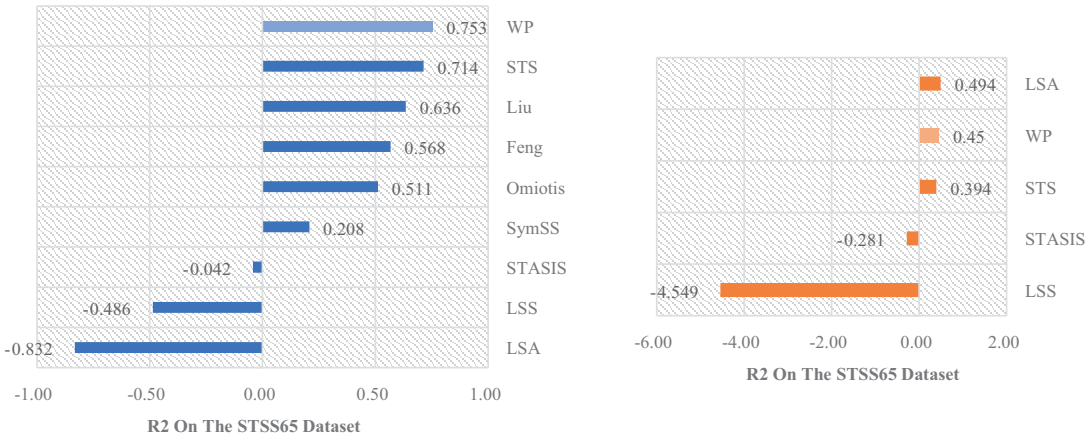

Figure 5. The comparison of our approach with recent methods in terms of R^2.

results. Therefore, extracting statistical features such as in STS and extracting good features from similarity matrices have been key to good results on this test.

On the STSS131 dataset, LSA, our approach, and STS were capable of producing better RSE. We reckon that because the WP and STS use more than one measure to compute similarity, they have produced small RSE values. LSA has been successful due to the possible high topic overlap between sentence pairs. LSS and STASIS follow a similar approach to produce relative sentence vector pairs. However, LSS has produced undesired results on this test.

3.5. R^2 Statistics

This is another metric to compute the similarity between the predicted regression and the true regression values. This measure can compute how much of the variability of the target regression line was explained by the predicted model and it is computed using Equation 8 [36]. The output of one means the perfect correlation of two regressions. The comparison values are given in Figure 5.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{8}$$

STS and our approach have outperformed other methods on this test. Our approach has been able to explain just over 2/3 of the variability of the target regression. LSA, LSS, and STASIS have obtained negative values which means that whenever the target value goes up, the predicted value goes down and the other way around. In general, our approach obtained the first or second best performance on all tests. On the STSS65, the STS has always been second best while the Feng, Liu, and SymSS methods have been mediocre on all tests. Overall, between all WordNet-based metrics (that is except LSA and STS), our method outperforms all the others on all the tests. This could be due to the efficient incorporation and feature extraction of WordNet-based similarities.

On the STSS131, once again LSA, WP, and STS performed best. STASIS produced poor results and LSS fits the data really poorly. Our approach (WP) outperforms the compared approaches on this dataset. The results of this section also show that only Pearson's or Spearman's correlation results are not sufficient for a fair comparison between different methods.

4. Conclusion

This study sets out to design a novel system for the similarity assessment of sentence pairs. This paper has argued that the recent approaches have not been thorough in feature extraction from similarity matrices and the importance of information content value was neglected in most of the studies. The novelties of this study are: the usage of YagoNet as the source of information for word pair similarity, the usage of geographical database and Wikipedia for name entity recognition and incorporating sentence vector similarity with the results of the similarity matrix. In this study, we have confirmed that the combination of weighted path metric and paraphrase database has improved the correlation results in comparison to other studies. This study has shown that extracting Yago-based features from the similarity matrix together with sentence vector similarity has improved the results of correlation in comparison to other studies on the STSS65 dataset. We have achieved the best correlation on the STSS65 and we have achieved comparable results on the STSS131 dataset. The results of this investigation show that our proposed approach performs better than all methods on many types of tests. Thus, the findings of this investigation complement those of earlier studies while these findings have significant implications for the understanding of the importance of feature extraction. The study has confirmed the findings in [6]; Huang et al. found that the information content value of the common concept is a very informative feature for similarity prediction. The most important limitation lies in the fact that phrasal verb was not considered in similarity for which a future work can be done to design a framework capable of assessing such cases in sentence pairs. Thus, further research needs to set identical words aside and compute similarity based on different words in sentence pairs.

References

[1] Mirhosseini M. A clustering approach using a combination of gravitational search algorithm and k-harmonic means and its application in text document clustering. Turkish Journal of Electrical Engineering & Computer Sciences 2017; 25 (2): 1251–1262. doi: 10.3906/elk-1508-31

[2] Güran A, Bayazit NG, Gürbüz MZ. Efficient feature integration with Wikipedia-based semantic feature extraction for Turkish text summarization. Turkish Journal of Electrical Engineering & Computer Sciences 2013; 21 (5): 1411–1425. doi: 10.3906/elk-1201-15

[3] Ur Rehman Khan S, Arshad Islam M, Aleem M, Azhar Iqbal M. Temporal specificity based text classification for information retrieval. Turkish Journal of Electrical Engineering & Computer Sciences 2018; 26 (6): 2916-2927. doi: 10.3906/elk-1711-136

[4] Ilgen B, Adali E, Tantuğ AC. Exploring feature sets for Turkish word sense disambiguation. Turkish Journal of Electrical Engineering & Computer Sciences 2016; 24 (5): 4391–4405. doi:10.3906/elk-1408-77

[5] Papineni K, Roukos S, Ward T, Zhu W-j. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics; Philadelphia, PA, USA; 2002. pp. 311-318.

[6] Huang H, Wu H, Wei X, Gao Y, Shi S. Mapping sentences to concept transferred space for semantic textual similarity. Knowledge and Information Systems 2018; 1-24. doi:10.1007/s10115-018-1261-3

[7] Deerwester S, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. Journal of the Association for Information Science and Technology 1990; 41 (6): 391–407. doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[8] Islam MA, Inkpen D. Second order co-occurrence PMI for determining the semantic similarity of words. In: Proceedings of LREC 2006; Genoa, Italy; 2006. pp. 1033-1038.

[9] Islam A, Inkpen D. Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data 2008; 2 (2): 1–25. doi:10.1145/1376815.1376819

[10] Cho JM, Seo J, Kim GC. Verb sense disambiguation based on dual distributional similarity. Natural Language Engineering 1999; 5 (2): 157-170. doi:10.1017/S1351324999002193

[11] Hirst G, St-Onge D. Wordnet: A Lexical database for English. In: Human Language Technology, Proceedings of a Workshop Held at Plainsboro, NJ, USA; 1994. pp. 468.

[12] Ghazizadeh Ahsaee M, Naghibzadeh M, Yasrebi Naeini SE. Semantic similarity assessment of words using weighted WordNet. International Journal of Machine Learning and Cybernetics 2014; 5 (3): 479-490. doi: 10.1007/s13042-012-0135-3

[13] Tsatsaronis G, Varlamis I, Vazirgiannis M. Text relatedness based on a word thesaurus. Journal of Artificial Intelligence Research 2010; 37: 1-39. doi: 10.1613/jair.2880

[14] Resnik P. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language Journal of Artificial Intelligence Research 1999; 11 (3398): 95-130. doi: 10.1613/jair.514

[15] Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S et al. The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations; Baltimore, MD, USA; 2014. pp. 55–60.

[16] Wu H, Huang H. Sentence similarity computational model based on information content. IEICE Transactions on Information and Systems 2016; E99.D (6): 1645-1652. doi:10.1587/transinf.2015EDP7474

[17] Wu H, Huang H. Efficient algorithm for sentence information content computing in semantic hierarchical network. IEICE Transactions on Information and Systems 2017; E100.D (1): 238-241. doi: 10.1587/transinf.2016EDL8177.

[18] Oliva J, Serrano JI, Del Castillo MD, Iglesias Á. SyMSS: A syntax-based measure for short-text semantic similarity. Data and Knowledge Engineering 2011; 70 (4): 390-405. doi: 10.1016/j.datak.2011.01.002

[19] Minkov E, Cohen WW. Adaptive graph walk-based similarity measures for parsed text. Natural Language Engineering 2014; 20 (3): 361-397. doi: 10.1017/S1351324912000393

[20] Iosif E, Potamianos A. Similarity computation using semantic networks created from web-harvested data. Natural Language Engineering 2015; 21 (1): 49-79. doi: 10.1017/S1351324913000144

[21] Sultan MA, Bethard S, Sumner T. DLSCU: sentence similarity from word alignment. In: Proceedings of the 8th International Workshop on Semantic Evaluation; Dublin, Ireland; 2014. pp. 241-246.

[22] Berndt J, Clifford D. Using dynamic time warping to find patterns in time series. In: AAAIWS'94 Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining; Seattle, WA, USA; 1994. pp. 359-370.

[23] Liu X, Zhou Y, Zheng R. Sentence similarity based on dynamic time warping. In: International Conference on Semantic Computing; Irvine, CA, USA; 2007. pp. 250-256.

[24] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 1970; 48 (3): 443-453. doi: 10.1016/0022-2836(70)90057-4

[25] Feng J, Zhou Y, Martin T. Sentence similarity based on relevance. In: Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008); Málaga, Spain; 2008. pp. 833-839.

[26] Croft D, Coupland S, Shell J, Brown S. A fast and efficient semantic short text similarity metric. In: 13th UK Workshop on Computational Intelligence UKCI; Guildford, United Kingdom; 2013. pp. 221-227.

[27] Li Y, McLean D, Bandar ZA, O'Shea JD, Crockett K. Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering 2006; 18 (8): 1138-1150. doi: 10.1109/TKDE.2006.130

[28] Zhu G, Iglesias CA. Sematch: Semantic similarity framework for Knowledge Graphs. Knowledge-Based Systems 2017; 130: 30-32. doi:10.1016/j.knosys.2017.05.021

[29] Zhu G, Iglesias CA. Computing Semantic Similarity of Concepts in Knowledge Graphs. IEEE Transactions on Knowledge and Data Engineering 2017; 29 (1): 72–85. doi: 10.1109/TKDE.2016.2610428

[30] Kuhn HW. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 1955; 2: 83-97. doi: 10.1007/978-3-540-68279-0_2

[31] Ganitkevitch J, Callison-Burch C. The multilingual paraphrase database. In: the 9th Edition of the Language Resources and Evaluation Conference. European Language Resources Association; Reykjavik, Iceland; 2014. pp. 4276-4283.

[32] Pavlick E, Rastogi P, Ganitkevitch J, Van Durme B, Callison-Burch C. PPDB 2.0: better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing; Beijing, China; 2015. pp. 425–430.

[33] O'Shea J, Bandar Z, Crockett K, McLean D. A comparative study of two short text semantic similarity measures. In: Nguyen NT, Jo GS, Howlett RJ, Jain LC (editors). Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Heidelberg: Springer, 2008, pp. 172-181.

[34] O'shea J, Bandar Z, Crockett K. A new benchmark dataset with production methodology for short text semantic similarity algorithms. ACM Transactions on Speech and Language Processing (TSLP) 2013; 10 (4): 1-63. doi: 10.1145/2537046

[35] Hauke J, Kossowski T. Comparison of values of Pearson's and spearman's correlation coefficients on the same sets of data. Quaestiones Geographicae 2011; 30 (2): 87-93. doi: 10.2478/v10117-011-0021-1

[36] James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning with Applications in R. New York, NY, USA: Springer-Verlag, 2013.