Journal of Theoretical and Applied Information Technology
15th December 2019. Vol.97. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A MULTI-LAYER SYSTEM FOR SEMANTIC RELATEDNESS EVALUATION

WAEL HASSAN GOMAA
Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef, Egypt
Shaqra University, Kingdom of Saudi Arabia
E-mail: wael.goma@gmail.com

ABSTRACT

Measuring semantic relatedness between sentences has always been a major point of discussion for NLP researchers. Semantic relatedness measures are key factors in text intelligence applications as paraphrase detection, short answer grading and information retrieval. This work highlights the effect of investing multiple similarity features by presenting a hybrid multi-layer system where each layer outputs a different independent similarity feature that are then merged using a simple machine learning model to predict text relatedness score. The system layers cover string-oriented, corpus-oriented, knowledge-oriented and sentences embeddings similarity measures. The proposed model has been tested on Sick data set that contains 9840 English sentence pairs. Experiments confirmed that using multiple similarity features is significantly better than applying each measure separately.

Keywords: *Semantic Relatedness, Sentence Embeddings, Text Similarity, Skip-Thought Vector, InferSent*

1. INTRODUCTION

Comparing text to detect semantic relatedness is a major concern in language processing tasks such as topic detection and tracking, text summarization, paraphrase detection, information retrieval, text clustering, document classification and machine translation. Starting with two sentences, a semantic relatedness system is expected to give a relatedness score (on a continuous scale) that represents how far the given sentences are related in terms of meaning. The real challenge behind measuring sentence similarity is the variability of linguistic expression and the limited amount of annotated training data [1].This research presents a semantic relatedness multi-layer model.

The proposed model includes four basic layers with different similarity techniques: string-oriented, corpus-oriented, knowledge-oriented and sentence embeddings. String-oriented similarity considers the order of characters and words. They output a score representing distance between two compared text strings [2].Corpus-oriented similarity measures take the advantage of information obtained from large corpora to judge text similarity. The huge amount of written or spoken texts available in the corpus positively affects the results of language research. Knowledge-oriented similarity calculates relatedness among terms using data stored in WordNet [3].Sentence embedding is a set of NLP techniques in which word sequences are converted to fixed-length vectors to determine similarity between sentences [4].The model presented in this research performs and evaluates each of the four similarity measures separately in a separate layer then the four layers are merged using a simple classical machine learning model. The suggested model has been tested on data set named Sick that contains 9840 pairs of sentences [5].Table 1 shows samples of sentence pairs together with the score of relatedness; obtained relatedness scores are ranged from 1 to 5 points.

The rest of this paper is organized as follows: section 2 navigates the semantic relatedness related work. An illustration for the suggested hybrid model is presented in section 3 followed by a listing for experiment results in section 4. Conclusion and suggested future work are presented in section 5.

2. RELATED WORK

Calculating semantic relatedness between words and sentences can be considered the biggest beneficiary of the non-stopping research in NLP. This section addresses some previous research to highlight the strengths and weakness of the previously used methods and illustrates the obstacles faced in computing semantic relatedness.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Relatedness Score Of Sample Pairs Of Sentences.

Sentence Pairs	Relatedness Score
1st Sentence: A boy is standing next to the opening of a fountain 2nd Sentence: The man is standing on a rocky mountain and gray clouds are in the background	1.4
1st Sentence: An officer is talking to the recruits 2nd Sentence: The recruits are barking at the military officer	2.885
1st Sentence: A chef is discarding some food 2nd Sentence: A chef is preparing a meal	3.2
1st Sentence: Some food is being prepared by a chef 2nd Sentence: A chef is preparing some food	4.815

In [6] five knowledge resources were introduced: WordNet, Wikipedia, Wiktionary, Web search engines and Semantic web. The research in [7] classified the methods of computing semantic relatedness into four major categories: Word Co-occurrence, Lexical Database, Deep Neural Network and Search Engine. Word co-occurrence approaches were widely used in NLP tasks. It determines the relationship between words by using a co-occurrence matrix that describes the frequency of words occurring together in a given corpus. The Co-occurrence matrix is decomposed into factors that when combined together they form the word vector representation [8]. The main drawbacks of this method is ignoring the word order of the sentence and bypassing the importance of the word meaning in the context of the sentence. On the other side this method is efficient in extracting keywords from documents and calculating similarity regardless the size of documents [7]. The second knowledge source is Lexical Database. A lexical database may contain lexical, syntactic, phonological and semantic relations. A well-known example of lexical databases is WordNet [2]. It contains many sematic relations such as synonymy, hyponymy and meronymy. Relatedness between two given words depends on both the path distance between words and the word information in the WordNet hierarchy. The third knowledge source is the enormous results of search engines. Passing a search query to a search engine will result a set of helpful information in form of text snippets. An application of this method is Google Similarity Distance (GSD) and Normalized Google Distance (NGD); these methods are based on the count of resulted web pages by Google search engine for any query after extracting the main keywords [9]. The fourth method is Deep Neural Network which proved noticeable improvements in semantic relatedness results. It adopts Word embedding (Word2vec) and Sentence embedding (Sent2vec); these methods convert word and sentence into fixed-length vector. The obtained vectors are captured from the syntactic and semantic relations between words and sentences. Sentence embeddings are categorized into specific task models and generic models [10]. In specific task models; experiments of certain task is trained and evaluated via applying different supervised classifiers. The most commonly used learning or training models in this area are the different types of neural networks such as LSTM, CNN and RNN. In generic models; semi supervised and unsupervised techniques are applied to get the sentences vectors. Various text-related tasks like semantic relatedness and sentiment analysis are depended on the generic models. There are many types of generic models such as Skip-thought vectors, GRAN and Infersent [11]. The idea of building multilayer system for improving semantic relatedness results was previously discussed in much research. One of these researches presented a pipeline system that includes four layers: word alignment, string-oriented, semantic-oriented and syntactic-based model [12].The system achieved promising accuracy results. Another research that worth mentioning is [13], which suggested a co-learning model that aimed at improving the performance of text entailment and relatedness tasks.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

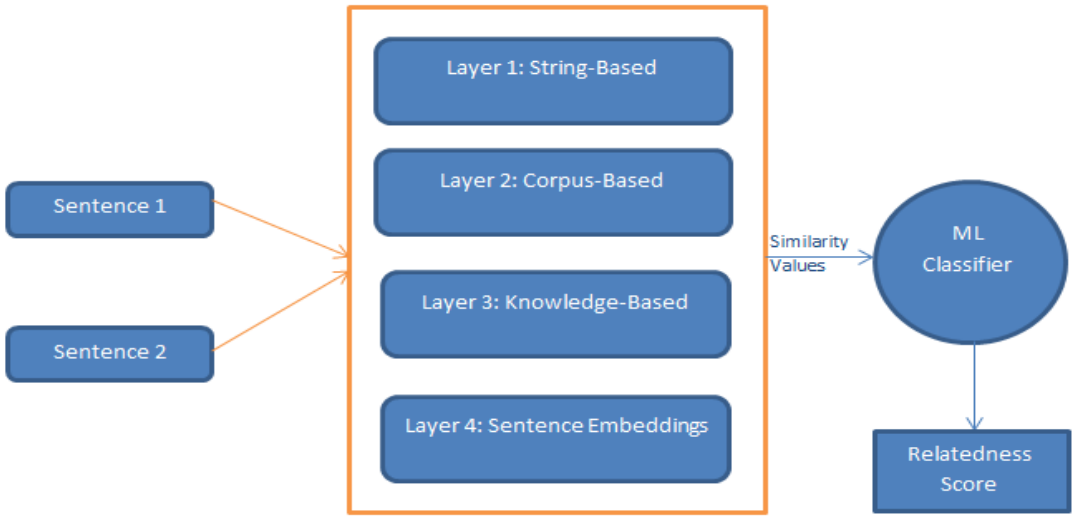

Figure 1: Proposed Model Architecture

The system didn't only use typical features like string-oriented similarity and N-Gram models, but also took the advantage of other features like the order of words, grammar structure and semantic-oriented similarity.

3. PROPOSED MODEL

Figure 1 shows overview of the suggested hybrid multi-layer system which includes four basic layers: String-oriented, Corpus-oriented, Knowledge-oriented and Sentences Embeddings. The extracted similarity features are then merged using a simple machine learning model to predict the final relatedness score. All text similarity algorithms in this paper are evaluated using SimAll package presented in [14]; SimAll provides a bunch of useful features as: availability of 61 different similarity algorithms in a single tool with an option of using them separately or in combination, upgrading similarities from word level to sentence level, supporting many languages as Arabic and English, covering five different pre-processing tasks and finally constructing different machine learning tasks. Figure 2 shows the text similarity approaches and algorithms that are applied in this paper.

3.1 Layer 1: String-Oriented Similarity

As described in section 1; string-oriented similarity is based on the order and alignment of characters and words. Authors in [2] classified string-oriented similarity into two categories: Character-oriented and Term-oriented. Character-oriented algorithms include Needleman Wunsch, Jaro, N-gram, Jaro Winkler, Damerau Levenshtein, Longest Common sub-string and Simth Waterman. Term-oriented algorithms include Dice, Euclidean, Block Distance, Overlap and matching coefficient, Cosine and Jaccard Similarity.

3.2 Layer 2: Corpus-Oriented Similarity

Corpus-oriented similarity approaches take the advantage of information obtained from large corpora to determine text similarity [2]. Four corpus-oriented algorithms are experimented in this research: LSA, LDA, DISCO 1 and DISCO 2. Starting with LSA; LSA stands for Latent Semantic Analysis. It works on the hypothesis that words near in meaning appear in similar pieces of text. Large pieces of text are scanned to build a matrix with number of words per paragraph, SVD matrix reduction method is applied before determining the similarity between rows using cosine similarity. LDA stands for Latent Dirichlet Allocation; it is a kind of unsupervised topic modeling algorithms. Documents in LDA are constructed over latent topics via random mixture, where each topic is defined by words distribution. DISCO stands for extracting DIstributionally related words using CO-occurrences. It determines apportionment similarity and relatedness among words by scanning and counting the occurrences words in a text window with different sizes. In cases where exact similarity between words is required; DISCO computes the degree of similarity by applying Lin algorithm. When the most similar distributional

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

word is required; DISCO simply returns the second order word vector. DISCO package contains two levels of similarity that cover both first order and second order relatedness.

3.3 Layer 3: Knowledge-Oriented Similarity

Knowledge-oriented Similarity determines semantic similarity and relatedness among terms using concepts and information obtained from semantic networks [2]. A well-known semantic network in Knowledge-oriented similarity measures is WordNet. Semantic similarity between terms is determined according to their likeness only. Semantic relatedness is a more comprehensive concept that is not restricted to the shape or form of the word. Knowledge-oriented algorithms include Res, Lin, JCN, LCH, WUP, Path, HSO, Lesk and Vector. Three algorithms are tested in this paper: Lin, Path and Lesk.

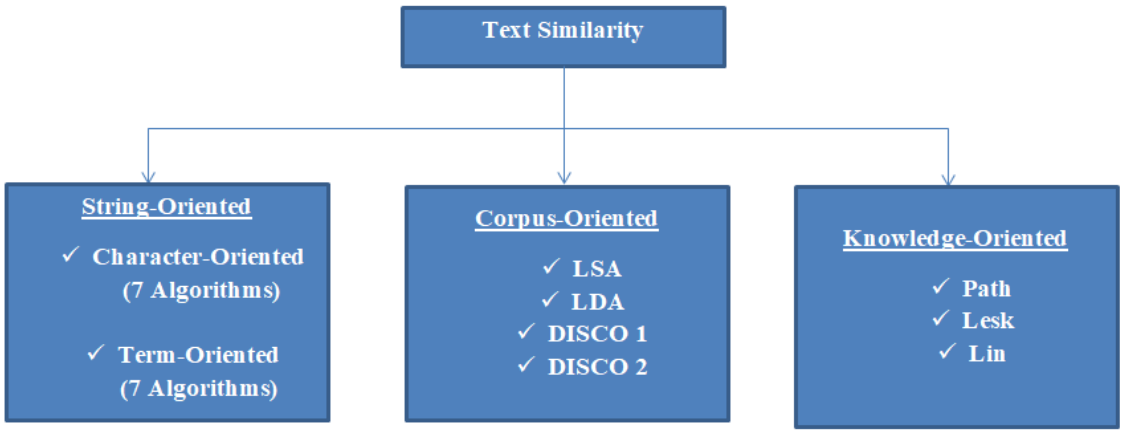

Figure 2: Text Similarity Approaches

3.4 Layer 4: Sentence Embedding

Neural networks proved to be an efficient tool in constructing representations of natural languages regardless the degree of linguistic abstraction. Word embeddings is a widely used representation in NLP. Actually it's not an exaggeration to say that Word embeddings are basic building blocks for NLP. While word embeddings capture similarities between words, research is paying more attention to compute embeddings that capture the semantics of word sequences (phrases, sentences, and paragraphs). Sentence embedding is the process of turning variable length text into numeric fixed-length vectors. Many methods are presented in this area ranging from simple additional composition of the word vectors to complicated structures as RNN and CNN. In this paper two sentence embedding models are examined: Skip-Thought Vector [15] and InferSent [16].

Skip-Thought vector model predicts sentences surrounding a given sentence instead of predicting words surrounding a word. As shown in figure 3, the model architecture is based on recurrent neural network; it has three parts: encoder network, previous decoder network and next decoder network [15]. The proposed model was tested on a 4800-dimentional combine-skip pre-trained model. Combine-skip model is resulted from merging unidirectional and bidirectional models.

InferSent is a sentence embeddings model that is trained on Natural Language Inference (NLI) data and can be efficiently utilized in many NLP tasks. Figure 4 explains the flow of InferSent model. Authors in [16] showed that a BiLSTM network with max pooling achieved the best current universal sentence encoding methods that even outperforms Skip Thought vectors. BiLSTM is a bi-directional LSTM network which computes n-vectors for n-words and each vector is a concatenation of outputs from a forward and a backward network that read the sentence in reverse direction. Then a max/mean pool is applied to each of the concatenated vectors to form the fixed-length final vector as shown in figure 5. In this paper, we tested two pre-trained model; InferSent1 that depends on FastText [17] pre-trained word vectors and infersent2 that depends on Glove [18] pre-trained word vectors. Both models output vectors are of dimension 4096.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

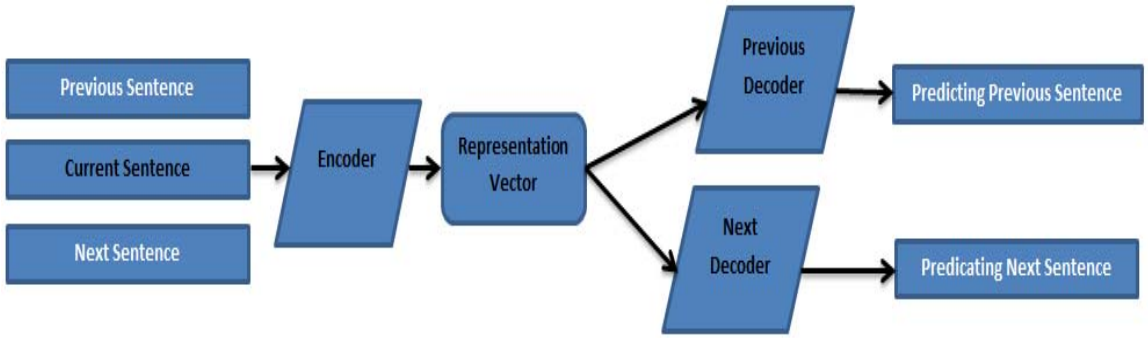

Figure 3: Skip-Thought Vector Architecture

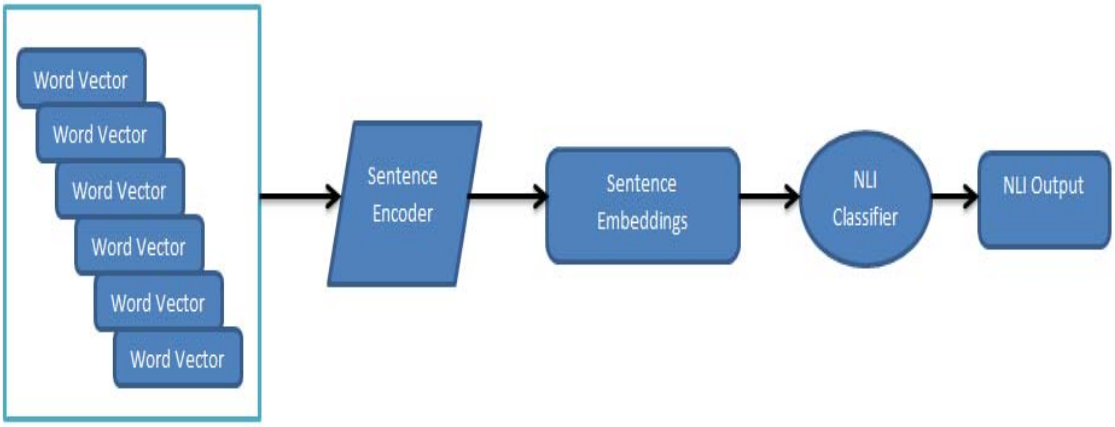

Figure 4: InferSent Flow

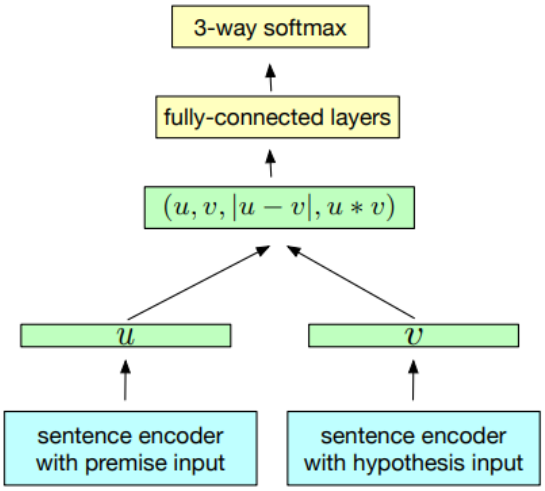

Figure 5: InferSent NLI Classifier [16]

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. RESULTS

This research presented a multi-layer architecture for semantic relatedness that operated on Sick data set which contains about 10,000 English sentence pairs. Table 2 and figure 6 state the result of all algorithms that applied separately in each layer previously explained in section 3. Results are obtained using Pearson correlation. The best correlation values of the layers in order are 0.625, 0.781, 0.771 and 0.888 resulting from N-gram, Lin, Disco2 and InferSent2 respectively. In overall, the results of sentence embeddings techniques showed significance of the overall performance.

TABLE 2: CORRELATION RESULTS FOR ALL LAYERS.

Algorithm	Pearson Correlation Coefficient
Layer 1: String-oriented Similarity	
Jaro	0.366
Jaro-Winkler	0.376
Needleman-Wunsch	0.454
Damerau-Levenshtein	0.474
Simth-Waterman	0.524
Euclidean distance	0.563
Longest Common SubString (LCS)	0.585
Jaccard Similarity	0.593
Matching Coefficient	0.594
Dice's Coefficient	0.602
Overlap Coefficient	0.602
Block Distance	0.613
Cosine Similarity	0.616
N-gram	**0.625**
Layer 2: Knowledge-oriented Similarity	
Path	0.754
Lesk	0.742
Lin	**0.781**
Layer 3: Corpus-oriented Similarity	
LSA	0.750
LDA	0.741
DISCO1	0.762
DISCO2	**0.771**
Layer 4: Sentence Embeddings	
Skip-Thought	0.858
InferSent1	0.884
InferSent2	**0.888**

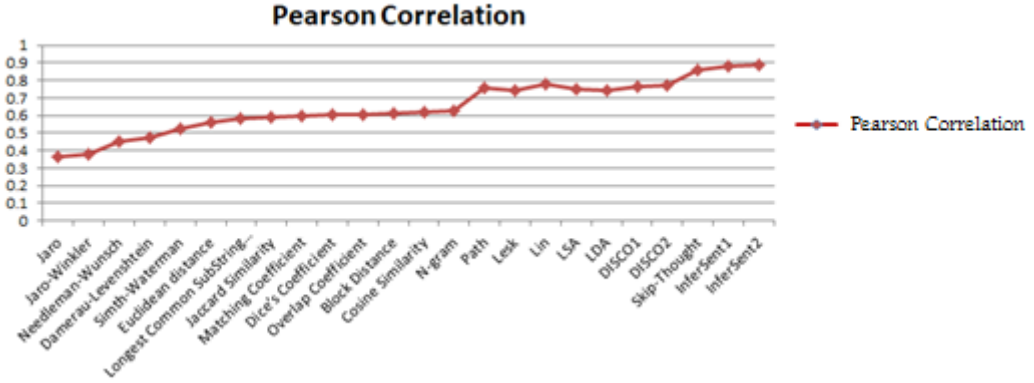

Figure 6: Correlation Results for All Algorithms

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Output similarity values from all layers were subjected to machine learning by using Weka. All experiments were validated using 10-fold cross-validation. MergeAll and MergeBest methods were used to train obtained similarity results [19].

MergeAll was trained on results of all of the 24 algorithms. MergeBest operated only on results of 8 algorithms : N-gram and Cosine similarity from layer 1, Lin from layer 2, DISCO1 and DISCO2 from layer 3, Skip-Thought, Infersent1 and InferSent2 from layer 4 .

These algorithms were selected by CfsSubsetEva attribute evaluator. CfsSubsetEva evaluates features according to their predictive ability and frequency of their occurrence together. It adopts Best First method to search sets of features by using greedy hill-climbing augmented with a backtracking facility.

Table 3 and figure 7 present the experiments results of the suggested merging methods along with Weka machine learning algorithms; which enhanced the accuracies obtained without applying the merging approach. Best correlation value was 0.903 obtained from the SMO classifier using MergeBest approach. SMO classifier normalizes all the given similarity values and uses pairwise classification technique. The proposed merging method consistently achieved accuracy which is very promising and exceeded previous results. To conclude the proposed work, experiments were performed through two stages. In the first stage, different similarity and relatedness algorithms are tested separately. The obtained results showed strong performance of sentences embeddings techniques for semantic relatedness task. On the other hand, we found that String-Oriented algorithms didn't perform well except for N-gram. Algorithms in both Corpus-Oriented and Knowledge-Oriented layers achieved promising results. The main idea of the second stage was to combine the important extracted similarity features from all the relatedness algorithms. Final results proved that the proposed hybrid model benefited from the strengths of all the used algorithms.

TABLE 2: CORRELATION RESULTS FOR ALL LAYERS.

Method	Correlation	Weka Classifier
MergeAll	0.891	NaiveBayes
MergeAll	0.892	BayesNet
MergeAll	0.894	Logistic
MergeBest	0.894	Logistic
MergeBest	0.893	SGD
MergeBest	0.896	MultilayerPerceptron
MergeBest	0.896	VotedPerceptron
MergeBest	**0.903**	**SMO**

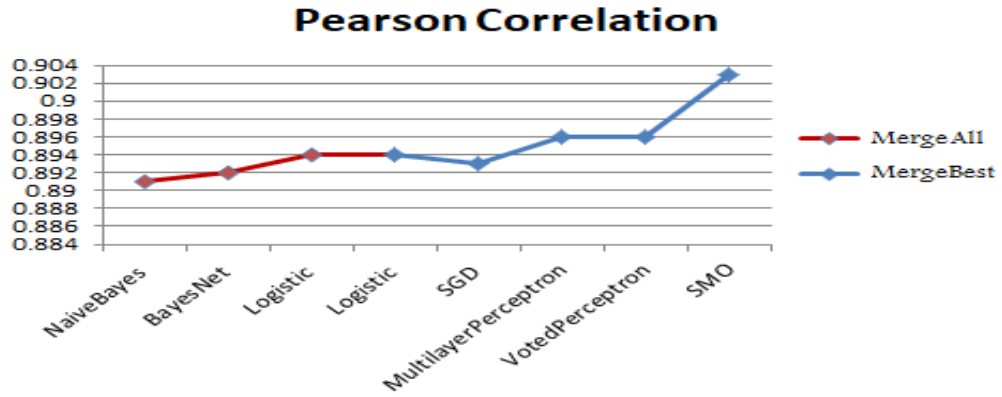

Figure 7: Correlation Results using MergeAll and MergeBest Methods

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSION AND FUTURE WORK

This paper presented a novel architecture to examine the challenge of semantic relatedness using Sick data. A four layered model of String-oriented, Corpus-Based, Knowledge-oriented and Sentence Embeddings was introduced. The framework utilized 24 different similarity algorithms that were evaluated separately and in combination. Two merging methods were used, MergeAll and MergeBest. MergeAll was trained on all the of the similarity values obtained separately from the 24 algorithm. MergeBest worked on only 8 algorithms that were automatically extracted using Weka machine learning algorithm. The proposed framework achieved promising correlation results. Future work includes testing the suggested architecture to Arabic language and experimenting other sentence embedding and deep learning transformation approaches.

REFRENCES:

[1] He, H., Gimpel, K., & Lin, J. (2015, September). Multi-perspective sentence similarity modeling with convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1576-1586).

[2] Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. International Journal of Computer Applications, 68(13), 13-18.

[3] Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).

[4] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196).

[5] Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., & Zamparelli, R. (2014). Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014) (pp. 1-8).

[6] Feng, Y., & Bagheri, E. (2017). Methods and resources for computing semantic relatedness. Encyclopedia with Semantic Computing and Robotic Intelligence, 1(01), 1630005.

[7] Pawar, A., & Mago, V. (2019). Challenging the boundaries of unsupervised learning for semantic similarity. IEEE Access, 7, 16291-16308.

[8] Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, 13(01), 157-169.

[9] Cilibrasi, R. L., & Vitanyi, P. M. (2007). The google similarity distance. IEEE Transactions on knowledge and data engineering, 19(3), 370-383.

[10] Jiao, X., Wang, F., & Feng, D. (2018, August). Convolutional neural network for universal sentence embeddings. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 2470-2481).

[11] Gomaa, W. H., & Fahmy, A. A. (2019, March). Ans2vec: A Scoring System for Short Answers. In International Conference on Advanced Machine Learning Technologies and Applications (pp. 586-595). Springer, Cham.

[12] Vo, N. P. A., & Popescu, O. (2016, November). A Multi-Layer System for Semantic Textual Similarity. In KDIR (pp. 56-67).

[13] Vo, N. P. A., & Popescu, O. (2016, November). Multi-layer and co-learning systems for semantic textual similarity, semantic relatedness and recognizing textual entailment. In International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management (pp. 54-77). Springer, Cham.

[14] Gomaa, W. H., & Fahmy, A. A. (2017, December). SimAll: A flexible tool for text similarity, In the Seventeenth Conference On Language Engineering ESOLEC' 2017 (pp. 122 127).

[15] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In Advances in neural information processing systems (pp. 3294-3302).

[16] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.

[17] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135-146.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[18] Pennington, J., Socher, R., & Manning, C. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[19] Gomaa, W. H., & Fahmy, A. A. (2014). Automatic scoring for answers to Arabic test questions. Computer Speech & Language, 28(4), 833-857.