Received: January 7, 2017

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Improving Semantic Textual Similarity with Phrase Entity Alignment

Vangapelli Sowmya[1]*, Bulusu Vishnu Vardhan[2], Mantena S.V.S. Bhadri Raju[3]

[1]*Gokaraju Rangaraju Institute of Engineering and Technology,Hyderabad, India*
[2]*Jawaharlal Nehru Technological University College of Engineering, Manthani, India*
[3]*Sagi Ram Ramakrishnam Raju Engineering College, Bhimavaram, India*
* Corresponding author’s Email: sowmyaakiran@gmail.com

Abstract: Semantic Textual Similarity (STS) measures the degree of semantic equivalence between two segments of text, even though the similar context is expressed using different words. The textual segments are word phrases, sentences, paragraphs or documents. The similarity can be measured using lexical, syntactic and semantic information embedded in the sentences. The STS task in SemEval workshop is viewed as a regression problem, where real-valued output is clipped to the range 0-5 on a sentence pair. In this paper, empirical evaluations are carried using lexical, syntactic and semantic features on STS 2016 dataset. A new syntactic feature, Phrase Entity Alignment (PEA) is proposed. A phrase entity is a conceptual unit in a sentence with a subject or an object and its describing words. PEA aligns phrase entities present in the sentences based on their similarity scores. STS score is measured by combing the similarity scores of all aligned phrase entities. The impact of PEA on semantic textual equivalence is depicted using Pearson correlation between system generated scores and the human annotations. The proposed system attains a mean score of 0.7454 using random forest regression model. The results indicate that the system using the lexical, syntactic and semantic features together with PEA feature perform comparably better than existing systems.

Keywords: Semantic textual similarity, Lexical, Syntactic, Semantic, Regression models, Phrase entity alignment, Pearson correlation coefficient.

1. Introduction

Semantic Textual Similarity (STS) is one of the core disciplines in Natural Language Processing (NLP). STS assess the degree of semantic similarity between two textual segments**.** The textual segments are phrases, sentences, paragraphs or documents. The objective of the STS is to measure the degree of equivalence in the range 0 to 5 between a sentence pair where 0 indicates both the sentences are on irrelevant topics and 5 indicates both the sentences mean the same thing [1]. STS system is trying to emulate the idea of similarity degrees, thus replicating human language understanding.

STS is related to both Textual Entailment (TE) and Paraphrase (PARA). STS is different from both TE and PARA. STS measures the graded semantic similarity whereas TE and PARA measures whether both the sentences are equivalent or not. STS measures the bidirectional graded equivalence whereas the TE measures the directional equivalence between two text segments. In many NLP applications STS is appropriately applicable than TE and PARA.

There are three approaches for measuring STS namely alignment based, vector space and machine learning [2]. Alignment approaches computes the similarity between the words or phrases in a sentence pair and aligns the words or phrases that are most similar, and then take the quality or coverage of alignments as similarity measure [3]. Vector space approach is a traditional NLP feature engineering approach represents the sentence as bag-of-words, and the similarity is evaluated according to the occurrence of words or co-occurrence of words or other replacement words [4]. Machine learning approaches uses supervised machine learning models to combine heterogeneous features such as lexical, syntactic and semantic features of sentence pair [5]**.**

However, estimating the semantic similarity between the sentence is difficult if both the sentences do not contain same words.

Potential applications of NLP such as Text summarization, Machine translation evaluation, Information retrieval, Web page retrieval, Plagiarism detection, Answer evaluation and Tweets search [6] can benefit from effective STS techniques.

In conventional approaches, there is no much work has done to identify word correlations with in a sentence to extract the syntactic information. In this paper, a syntactic feature PEA is proposed to address this issue. The STS score is produced by combining the lexical, syntactic and semantic features along with the proposed feature PEA through regression technique. The proposed system is evaluated using Pearson correlation coefficient between the manually annotated values and the system generated values. This system performs with a better score when compared to existing systems.

This paper is organized in nine sections. The related work in STS is described in section 2. The STS model is explained in section 3. Lexical and syntactic features are discussed in section 4. Semantic features are discussed in section 5. The proposed syntactic feature PEA is explained in section 6. The experimental work for pre-processing the data and model building is discussed in section 7. In section 8, the attainment of the results is depicted. Section 9, concludes this work with future possible extensions to the proposed work.

2. Literature Survey

The vast amount of literature has been done for measuring the similarity between long texts such as text documents [7] and less amount of work is done for measuring the similarity between short texts such as sentences or phrases [8]. The methods for measuring the similarity among texts is classified into vector based, corpus based, hybrid and feature based methods. Vector based models are used in Information retrieval systems [9]. The corpus based methods include Latent Semantic Analysis (LSA) [8] is a method for extracting and representing the contextual meaning of text by analyzing the large natural language text corpus. Hybrid methods involves corpus-based [10] and knowledge-based measures [11].

Feature-based methods represents a sentence by generating a set of features using syntactic and semantic information embedded in the sentence. The primary and composite features are introduced to build the feature vector of a text [7]. Primary features compare individual items of a text unit. Composite features are formed by combining two or more primary features. The challenging task in this method is finding the effective features that aids in measuring the semantic similarity and a classifier is required to build the model upon these features. Mihalcea et.al., [12] has proposed two corpus base measures and six knowledge based measures for finding the semantic similarity between word and a method which combines the information extracted from the similarity of component words to compute semantic similarity between two texts. Li et.al. [13] has proposed an unsupervised method which computes the similarity between two texts by combining both syntactic and semantic information. For obtaining the syntactic information the measure used is word order and for syntactic information is measured with the aid of knowledge-base and corpus-base. Islam et. al [14] proposed a method that measures the similarity between two texts by normalizing three features string similarity, common-word order and semantic similarity. The first two features string similarity and common-word order similarity emphasis on syntactic information whereas the semantic similarity emphasis on semantic information and it is calculated using corpus statistics. These methods mostly concentrated to identify semantic similarities among the words using knowledge and corpus based features. Some other methods are focused to identify more number of features instead of establishing syntactic relationships among the terms present in the sentences.

The STS task is annually conducted from 2012 till date for evaluating the newly proposed algorithms and models. The datasets MSRpar, MSRvid, OnWN, SMTnews, SMTeuroparl used in SemEval 2012 for evaluating the systems. The outperformed STS system in SemEval 2012 used Explicit Semantic Analysis (ESA)[15] and lexical similarity with a mean correlation coefficient of 0.6773 and scored highest correlation coefficient for MSRpar and MSRvid datasets. For OnWN dataset, the best performed system Weiwei [16] used simple unsupervised latent semantics based approach, Weighted Textual Matrix Factorization which uses bag-of-words features. This system is superior than LSA and Latent Dirichlet Allocation (LDA) because it handles the missing words in the sentence. For SMTeuroparl dataset, the maximum correlation attained was 0.5666 by the system sranjans [17] which graded the similarity between two sentences by finding maximal weighted bipartite match between the tokens of the two sentences.

The outperformed systems in SemEval 2012 are mostly concentrated on establishing semantic relations among the terms based on the corpus. The importance of lexical, syntactic relationships and knowledge base features using WordNet has not been considered.

In SemEval 2013 the dataset contains four different collections of data that includes HDL, FNWN, OnWN and SMT. The best model UMBC EBIQUITY-CORE [18] used LSA [19], Knowledge-source (WordNet) and n-gram matching techniques for finding the degree of equivalence between two sentences which scored a mean correlation of 0.6181 and achieved highest correlation for the datasets HDL and FNWN. The highest correlation 0.8431 for OnWN dataset has achieved by deft system which is based on distributional similarity. The system NTNU-CORE [20] used TakeLab features, DKPro features in addition with GateWordMatch feature and trained the system using Support Vector Regression(SVR) which attained maximum correlation 0.4035 for SMT dataset.

SemEval 2014 contains HDL, OnWN, Deft-forum, Deft-news, Images and Tweet-news datasets. The outperformed system is DLS@CU [21] with mean correlation 0.761, has aligned the related words in two sentences for measuring the semantic equivalent between two sentences. The system Meerkat Mafia [22] which is an unsupervised system used word-similarity model for term alignment has attained a correlation of 0.785 for deft-news. The system Meerkat Mafia [22] which is a supervised system to combine the scores generated from unsupervised system and the enhanced word-similarity wrapper has attained correlation of 0.779 ,0.763 and 0.875 for the Tweet-news, OnWN and HDL respectively. The system NTNU [23] combines measures using bagged support vector regression based on lexical soft cardinality and character n-gram feature representations with lexical distance metrics from TakeLab's baseline system which has attained correlation of 0.792 for Tweet-news, 0.834 for images and 0.53 for deft-forum datasets.

In SemEval 2013 and 2014, the importance of word order in its syntactic information has not addressed by any of the outperforming systems.

The datasets in SemEval 2015 are HDL, Images, Ans-student, Ans-forum and Belief. The best overall performance is achieved by DLS@CU [24] supervised system, which attains a mean correlation of 0.8015. Two systems are built one is unsupervised system which is based on word alignments between two input sentences and the other is an unsupervised system which uses word alignments and similarities between compositional sentence vectors as its features. The unsupervised DLS@CU [24] system has attained a correlation 0.7879 for answer-student dataset and the supervised DLS@CU [24] system has attained correlation 0.7390 for Ans-forum dataset. For Belief dataset IITNLP system has attained the highest correlation 0.7717. The system Samsung [25] improves the UMBC-Pairing Words system by semantically differentiating distributional similar terms, which attains correlation of 0.8417 and 0.8713 for headlines and images respectively.

The outperforming system in SemEval 2016 is built by Samsung_Poland_NLP_Team [26] with the highest correlation of 0.77807. The system uses an ensemble classifier, combining an aligner with a bi-directional Gated Recurrent Neural Network and RAE with WordNet features. It also attained a highest correlation 0.6923, 0.8274 and 0.8413 for the Ans-Ans, HDL and plagiarism dataset respectively. An unsupervised system MayoNLP [27] has attained 0.74705 for Ques-Ques dataset, which is built by combining linearly a feature which is based on lexical semantic nets with another feature based on deep learning semantic model. The RICOH [28] system has attained correlation of 0.8669, which is an IR based system that extends a conventional IR-based scheme by incorporating word alignment information.

In SemEval 2015 and 2016, the syntactic information is carried using word order and word alignment. The identification of phrase entities and the relationship among the phrase entities using knowledge and corpus base has not been addressed. In the present work, syntactic information in the form of phrases is identified, thus the STS score has improved significantly on the SemEval 2016 dataset.

3. Semantic textual similarity model

The objective of STS model in Fig. 1 is to measure the degree of equivalence in the range [0,5] between a sentence pair, where 0 indicates both the sentences are irrelevant, 1 indicates both the sentences are not equivalent but discussing about the same topic, 2 indicates both the sentences are not equivalent but share some details, 3 indicates both the sentences are roughly equivalent but important information is missing or differed, 4 indicates both the sentences are mostly equivalent as but some unimportant information differs and 5 indicates both the sentences are completely equivalent.

The proposed STS system is a supervised system. The SemEval 2016 dataset is considered for experimental evaluations. The data set contains set of sentence pairs with human annotated values. The dataset is divided into two disjoint training and testing sets.

Data is pre-processed in pre-processing stage using various pre-processing techniques. Pre-processing stage contains three steps such as contractions replacement, Lower case conversion and spelling corrections as shown in Fig. 2. The text contractions are replaced with full text. Lower case conversion is used for standardizing the text in checking the string equivalence and part of speech tagging. Misspelled and wrongly spelled words are corrected.

Various techniques are used to explore syntactic and semantic information embedded in the sentence pairs. Numerous Regression techniques exists to combine set of syntactic and semantic information to build a learned model. The learned model is used to find the degree of semantic equivalence between sentence pairs of the test dataset. The performance of learned model is measured using Pearson correlation between system generated and human annotated values of each sentence pair.

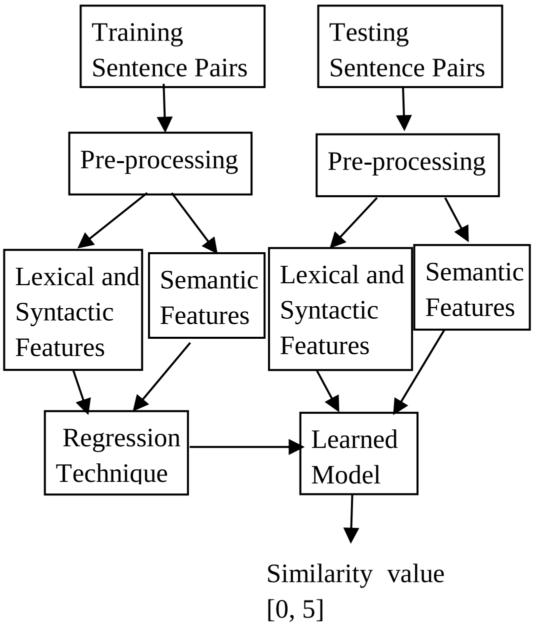

Figure.1 Semantic textual similarity system

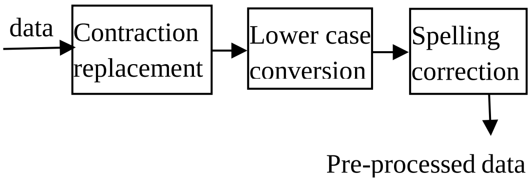

Figure.2 Pre-processing

4. Lexical and syntactic features

Lexical and syntactic features are used to measure the structure and stylistic similarity between a sentence pair. The various lexical and syntactic features are:

4.1 Set features

Set features of sentence pairs is captured using the cardinality of a set. The set of a sentence is defined as the unique words present in that sentence. For example, X and Y are two sets which contain the unique words present in sentence S1 and S2 respectively. Where S1 and S2 forms a sentence pair. 8 features: $|X|$, $|Y|$, $|X-Y|$, $|X-Y|/|Y|$, $|Y-X|$, $|Y-X|/|X|$ $|X \cap Y|$, $|XUY|$ are generated.

4.2 Longest Common Sequence(LCS)

LCS is the ratio between the length of the longest common word sequence (lcws) and the length of the shorter sentence.

$$LCS(s_1, s_2) = \frac{length(lcws(s_1, s_2))}{min(length(s_1), length(s_2))} \tag{1}$$

where, s_1 and s_2 represents sentence1 and sentence2 respectively.

4.3 N-gram features

n-grams at character, word, part-of-speech (pos) and lemmatize word level are generated. For character n-grams n = {2,3,4} and for word n-grams, pos n-grams, lemma n-grams n = {1,2,3}. The Jaccard similarity between these n-grams is calculated as

$$J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{2}$$

where *'X'* and *'Y'* are the set of corresponding n-grams.

4.4 Word order similarity

Word Order Similarity (WOS) is measured between two sentence vectors as shown in Fig. 3. Unique Word Vector(UWV) is formed from the unique words contained in the sentence pair. Alignment between the UWV and the words in the sentence is made with WordNet. UWV is constructed to make sentence vector1(SV_1) and sentence vector2(SV_2) of equal dimensions. To form SV_1, if the unique word(UW_i) of UWV is present in sentence1 at jth position then the ith entry of SV1 is the value ‘j’. Otherwise the similarity between unique word(UW$_i$) and all the words(W$_j$) in sentence1 is calculated from WordNet using S(UW$_i$, W$_j$) in Eq. 3. Then most similar word position is assigned to the ith entry in SV$_1$.

$$S\left(UW_i,\ W_j\right) = e^{\alpha l} \times \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \qquad (3)$$

where, ‘*l’* is the shortest path and *‘h’* is the depth measure between the words *UW$_i$* and *W$_j$* in the WordNet. *‘α’*, *‘β’* are the constants. The *‘α’* value and *‘β’* values are 0.2 and 0.45 respectively which are found to be best by Li[13].

The WOS between two sentence vectors is calculated as in Eq. (4).

$$WOS\ \left(SV_1,\ SV_2\right)\ =\ 1 - \frac{|SV_1 - SV_2|}{|SV_1 + SV_2|} \qquad (4)$$

where, *SV$_1$* and *SV$_2$* are sentence vector1 and sentence vector2 respectively.

4.5 Tf-idf

To form tf-idf vectors, UWV of a sentence pair is generated. From UWV the words with length equal to one are deleted. The UWV is sorted in dictionary order. For each sentence in the sentence pair a tf-idf vector is generated by calculating tf-idf(d, t) as in Eq. (6). The inverse document frequency idf(d,t) is calculated as

$$idf\,(d,t) = log_e\left(\frac{1+n}{1+df\,(d,t)}\right)+1 \qquad (5)$$

where, *‘t’* refers to term in UWV, *‘d’* refers to document, *‘n’* refers number of documents, ‘*df(d,t)’* is the document frequency i.e., the number of documents in which term *‘t’* is present.

As inverse document frequency (idf) is generated between two sentences, every sentence is considered as an individual document.

$$tf - idf\,(d,t) = tf \times idf\,(d,t) \qquad (6)$$

where, *‘tf’* is the term frequency i.e., the number of times term *‘t’* is present in the document *‘d’*.

Euclidean normalization is used to normalize the tf-idf vector (*v*).

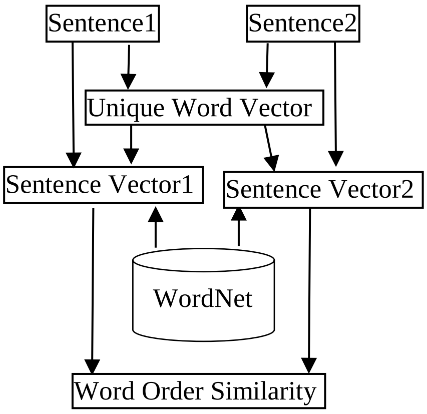

Figure. 3 Word order similarity architecture

$$Eucledian\_norm(v) = \frac{v}{|v|} \qquad (7)$$

After normalization, the cosine value between these two vectors is computed to find the similarity between two sentences.

5. Semantic features

Semantic features deal with the meaning of the words in the sentences.

5.1 Knowledge and corpus based feature

For finding the semantic similarity between a sentence pair, each sentence is mapped to the unique word vector(UWV) to form semantic vectors. If the unique word (UW$_i$) is present in the sentence then the ith entry in the semantic vector(SV) is 1 otherwise ith entry in the semantic vector(SV) is the highest semantic similarity value computed between the UW$_i$ and every word in the sentence using S(UW$_i$, W$_j$). For computing S(UW$_i$, W$_j$) lexical database is used i.e., WordNet. The information content values are calculated by incorporating corpus statistic. The corpus statistics is incorporated to make this feature work for various domains Li [13]. The value at the ith entry in semantic vectors generated using knowledge based feature i.e., S(UW$_i$, W$_j$) is normalized by multiplying S(UW$_i$, W$_j$) with IC(UW$_i$) and IC(W$_j$) to generate normalized semantic vectors (NSV). The IC(w) is the information content of word and it is defined as:

$$IC(w) = 1 - \frac{\log(n+1)}{\log(N+1)} \qquad (8)$$

The total number of words in the corpus is *'N'*. The term frequency of the word *'w'* in the corpus is indicated by *'n'*.

The semantic with corpus similarity is the cosine value between the normalized semantic vectors of the sentence pair.

5.2 Sent2Vec feature

Sent2Vec is used to generate feature vectors for a sentence pair with deep structured semantic model(DSSM) [29]. Similarly, Convolutional pooling deep structured semantic model (CDSSM) [30] is also used to generate feature vectors of a sentence pair. The similarity between the sentence pair is computed by finding the cosine similarity between the feature vectors.

6. Proposed syntactic feature: Phrase Entity Alignment (PEA)

A phrase entity is a conceptual unit in a sentence with a subject or an object and its describing words. The proposed syntactic feature, PEA initially identifies the phrase entities present in each sentence. Secondly, the semantic similarity score between two phrase entities is calculated using the knowledge and corpus based feature on words as in Li[13]. The phrase entities present in one sentence are aligned with the phrase entities present in other sentence based on their maximum semantic similarity score between them. Finally, the STS between two sentences is measured by combining the semantic similarity scores of all aligned phrase entities.

A phrase entity is formed with zero or one determiner, zero or more adjectives and a noun. The semantic similarity is computed between each pair of sentence phrase entities using WordNet and brown corpus Li [13]. The procedure for aligning the phrase entities and for computing the similarity between two phrase entities is as follows:

```
Algorithm PEAlign_Sim(pe_matrix, m, n)
Phrase Entity matrix pe_matrix, size of the matrix m,n
begin
  sim ← 0
  while pe_matrix is not empty do
      find i, j of maximum element e in pe_matrix
      add e to sim
      delete i^th row and j^th column from pe_matrix
  end while
  pe_sim ← sim/max(m,n)
  return pe_sim
end
```

A phrase entity matrix (pe_matrix) of dimensions m×n is constructed with the semantic similarity values between phrase entities where m and n represents the number of phrase entities in the first sentence and second sentence accordingly. Identify the maximum value 'e' from the pe_matrix that indicates the most similar phrase entities from two sentences. Then these two phrase entities are aligned. The similarity value (sim) is updated with the maximum value 'e'. Then the corresponding row and column of the maximum value are removed from the pe_matrix. The process is repeated until all phrase entities from these two sentences are aligned. The overall phrase entity similarity (pe_sim) between two sentences is calculated as the ratio between similarity value and to the maximum number of phrase entities in the sentence pair. The following example demonstrate the procedure to calculate the similarity value between two phrase entities.

s1: *a little yellow dog jumping on a black cat.*
s2: *a yellow dog jumping on a shiny black kitten.*
POS tagging:
s1: [[('a', 'DT'), ('little', 'JJ'), ('yellow', 'JJ'), ('dog', 'NN')], [('a', 'DT'), ('black', 'JJ'), ('cat.', 'NN')]]
s2: [[('a', 'DT'), ('yellow', 'JJ'), ('dog', 'NN')], [('a', 'DT'), ('shiny','JJ'), ('black', 'JJ'), ('kitten.', 'NN')]]
In s1, there are two phrase entities:
PE11: a little yellow dog
PE12: a black cat
In s2, there are two phrase entities:
PE21: a yellow dog
PE22: a shiny black kitten

where, PE11, PE12 are the phrase entities in sentence1 and PE21, PE22 are the phrase entities in sentence2.

pe_matrix:

	PE11	PE12
PE21	**0.9475**	0.2476
PE22	0.3829	**0.2514**

sim = **0.9475+0.2514=1.1989**
pe_sim=0.59945

7. Experiments

7.1 Dataset

The dataset contains 5 distinct categories of data that belongs to various domains such as headlines, plagiarism, question-question, answer-answer and post editing. The dataset contains sentence pairs with human annotated continuous values ranging from 0 to 5. The training dataset is collected from the previous SemEval workshops and the test dataset is collected from SemEval 2016 as presented in the Table 1.

Table 1. Mapping of Test set with training sets

Test Set	No. of test pairs	Training Sets	No.of training Pairs
answer-answer	254	answer_ students 2015	750
		belief 2015	375
headlines, plagiarism	249, 230	MSRpar 2012	1500
		SMTnews 2012	750
		deft_news 2014	300
		headlines 2013	750
		headlines 2014	750
		headlines 2015	750
		images 2014	750
		images 2015	750
Postediting	244	deft_news 2014	300
		deft_forum 2014	450
		SMTnews 2012	750
question_ question	209	deft_news 2014	300
		deft_forum 2014	450
		belief 2015	375

7.2 Pre-processing

The dataset is preprocessed before building the model to generate the features correctly as shown in Figure 2.

For question_question dataset stop words are removed because the similarity between them depends on content words rather than stop words. For the following example the human annotation is 0.
s1: *what is the best way to repair a cracked bathtub?*
s2: *what is the best way to clean a grater?*
After removing stop words
s1: *best way repair cracked bathtub?*
s2: *best way clean grater?*

7.3 Feature generation

Lexical, syntactic and semantic features for each sentence pair in the dataset is generated. There are 27 features generated in which 24 are lexical and syntactic features and the remaining are semantic features. In 24 lexical and syntactic features, there are 8 set features,1 lcs, 12 n-gram features out of which 3-character n-gram, 3 word n-gram, 3 pos n-gram and 3 lemma n-gram features, 1 word order similarity, 1 tf-idf and 1 phrase entity alignment feature. The 3 semantic features are knowledge and Corpus based feature and 2 sen2vec features. The set features are generated by dividing the sentence into tokens using NLTK tokenizer.

The set 'S' of a sentence contains the unique tokens present in that sentence. 8 set features are generated as discussed in 4.1. Character n-grams, word n-grams, pos n-grams are generated by first tagging part-of-speech by using nltk and lemma n-grams are generated using WordNetLemmatizer and ngram packages from nltk. The list of characters, words and lemmas are stored as per the order present in the sentences to preserve syntactic structure. Jaccard similarity is computed between the corresponding n-grams generated for a sentence pair. For finding the word order similarity lexical database WordNet is used to align the words in the sentence pair. tf-idf vector is generated by using the TfidfVectorizer from Sklearn and the cosine similarity between the vectors are computed to generate tf-idf feature value. For generating the phrase entity alignment feature each sentence in the sentence pair is tagged using the part-of-speech tagging from nltk then the sentence is divided into phrase entities. The phrase entities are aligned with the help of WordNet database and brown corpus as explained in section 6. Sent2Vec tool is used to generate both dssm and cdssm features.

7.4 Model building

The syntactic and semantic features are combined using regression models such as Support vector machine [31] and using various ensembling methods such as random forest, bagging and boosting [32].

The regression algorithms are used to build the model as the degree of semantic similarity is a continuous value scaled from 0 to 5. The importance of lexical, syntactic and semantic features, the influence of the proposed syntactic feature 'phrase entity alignment' on semantic textual similarity are evaluated by building a learnt model using various regression techniques. All the features discussed in section 4,5,6 are used as input to build the models for all the datasets except plagiarism dataset. From the experimental results, it is observed that the plagiarism dataset mainly depends on syntactic features. So, for the plagiarism dataset the Sent2Vec features are not considered. All the regression models are discussed in section 7 are implemented in R environment.

7.5 Model Evaluation

For model evaluation, the testing set sentence pairs are pre-processed then the features are generated and these features are given as input to the model built to generate the degree of semantic similarity value. The model outputs the continuous value from 0 to 5. For evaluating the model Pearson correlation coefficient is used as evaluation measure.

Pearson correlation coefficient is used to measure the relationship between two continuous valued variables. The value will range from -1 to 1, where [-1,0) indicates the variables are negatively correlated, 0 indicates both the variables are independent and (0,1] indicates they are positively correlated. The value approaching to 1 indicates the positive correlation is increasing between two variables. 1 indicates they are perfectly correlated. The Pearson correlation coefficient *'r'* is calculated between two variables *'x'* and *'y'* as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{9}$$

where x_i and y_i represents i^{th} value in vectors x and y respe4ectively, *n* represents the number of values in the vector, $\bar{x}$ and $\bar{y}$ are the mean values of *x* and *y* vectors respectively.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{10}$$

8. Results and discussions

For all the five datasets in SemEval 2016 corpus, it is observed that the PEA feature has improved the correlation between system generated values and human annotations. The results in Table 2 depict the importance of syntactic features and influence of PEA feature to calculate the degree of semantic textual similarity. To show the influence of PEA feature, the experiments are carried with the combination of PEA and all other syntactic features.

From the results, it is observed that the PEA feature improves the correlation between system generated similarity value and human annotated value. The examples presented below shows the influence of PEA on sentence pairs.

S1: *There are two things to consider.*

S2: *A couple of things to consider.*

The human annotated similarity value is 4 for this sentence pair. The similarity value between these two sentences using all lexical and syntactic features without (WO) including PEA is 3.96 whereas with (WI) including PEA with all other features is 4.10.

From the results depicting in Table 3, the model is built using semantic features in combination with PEA feature. The similarity value obtained with only semantic features for sentence pair is 3.12 and the value 3.3 is obtained when PEA feature is added to semantic features.

The model is built using all syntactic and semantic features with the combination of PEA feature. The results in Table 4 shows that PEA improves the mean Pearson correlation for all data sets. The similarity value between the sentences is 3.75 when the system is built using lexical, syntactic and semantic features. The similarity value 4.3 when the system is built with all the features in addition with PEA which is close to the human annotated value.

Set features establishes the relationship between the sentences based on their word count. It fails to address the relationship between the words. Word order finds the similarity between the sentences by identifying the position of a word in a sentence but does not consider the context of a word. LCS identifies only the longest common word matching sequence but fails to identify the concepts in a sentence. All n-gram features are useful in identifying set of sequence of terms with fixed length that present in two sentences, but not considering their syntactic relationships. Tf-idf calculates the importance of a word within a sentence and among the two sentences but does not establish syntactic relationship with the coexisting words in a sentence.

Semantic feature with knowledge and corpus based is useful to establish the relationship between the words based on their meaning even though the words are lexically inequivalent. But it fails to identify the context of a word within a sentence. Sent2Vec forms a vector based on the linguistic context of the words presented in the sentence. But in STS task, it is required to identify the semantic similarity of the concepts that are present in two sentences with multiple words having same meaning.

A sentence can have multiple concepts. A concept can be defined with varying number of words and with different words having same meaning. Semantic equivalence between sentences can be measured by identifying the concepts present in the sentences. All lexical, syntactic and semantic features that exists does not identify the concepts that are present in the sentences. PEA identifies and aligns the concepts having maximum semantic similarity.

To find the similarity between the documents cosine similarity is computed between the tf-idf vectors of the documents. So, the results of tf-idf is also depicted in Table 5. The baseline system is built using one-hot coding. In one-hot coding each sentence is represented as a vector. The vectors are built by using different words present in the two sentences. If a word in the vector is contained in the sentence then the value in the vector is 1 otherwise 0.

For measuring the similarity between two sentences the cosine similarity between two sentence vectors is calculated. The comparison is done between tf-idf, baseline system, UWB system which is one of the top performed system in SemEval 2016 and the models built using lexical and syntactic features with PEA and combined features with PEA in Table 5. The results show that the correlation is improved when the model is built with lexical, syntactic, semantic and PEA feature. For question-question dataset the baseline system and the tf-idf system performed too low because these two systems work on lexical overlap features. But the sematic textual similarity between the two questions depends on the meaning of the content words rather than the non-content words. Therefore, the results in the Table 5 for question-question depicts that semantic features have more influence than the other features. The plagiarism dataset contains lexical overlaps in the sentence and the syntactic structure of the sentence. So, the lexical and syntactic features have more influence for the plagiarism dataset.

Table 2. Pearson correlations on monolingual STS task of SemEval 2016 with lexical and Syntactic Features

Datasets	**Answer-Answer**		**Headlines**		**Plagiarism**		**Post-editing**		**Question-Question**	
Regression Models	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**
Random Forest	0.5862	**0.6043**	0.7304	**0.7522**	0.8099	**0.8111**	0.7739	**0.7761**	0.5700	**0.5809**
Bagging	0.6198	**0.6320**	0.7180	**0.7184**	0.8241	**0.8252**	0.7693	**0.7907**	0.5855	**0.5906**
Boosting	0.6154	**0.6188**	0.7478	**0.7543**	0.7527	**0.7650**	0.8381	**0.8408**	0.5757	**0.5790**
SVM	0.4855	0.4777	0.6744	**0.6813**	0.7039	0.6917	0.6729	**0.6996**	0.4813	**0.4836**

Table 3. Pearson correlations on monolingual STS task of SemEval 2016 with semantic and Phrase Entity Alignment Features

Datasets	**Answer-Answer**		**Headlines**		**Plagiarism**		**Post-editing**		**Question-Question**	
Regression Models	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**	**WO PEA**	**WI PEA**
Random Forest	0.5716	0.5876	0.7090	**0.7196**	0.6722	**0.6763**	0.8019	**0.8031**	0.7198	0.6833
Bagging	0.5671	**0.5989**	0.7267	0.7267	0.7222	0.7208	0.8204	0.8171	0.7111	0.7103
Boosting	0.6221	**0.6372**	0.7598	**0.7601**	0.7413	**0.7510**	0.8317	**0.8386**	0.7150	0.7147
SVM	0.5723	0.5653	0.7509	0.7494	0.7265	0.7250	0.8134	**0.8193**	0.7214	0.7032

Table 4. Pearson correlations on monolingual STS task of SemEval 2016 with Lexical, Syntactic and Semantic Features

Datasets / Regression Models	Answer-Answer		Headlines		Plagiarism		Post-editing		Question-Question		Mean	
	WO PEA	WI PEA	WO PEA	WI PEA	WO PEA	WI PEA	WO PEA	WI PEA	WO PEA	WI PEA	WO PEA	WI PEA
Random Forest	0.6143	**0.6220**	0.7778	**0.7847**	0.7988	0.7981	0.8064	**0.8156**	0.6942	**0.7068**	0.7383	**0.7454**
Bagging	0.6472	**0.6528**	0.7395	0.7395	0.8067	0.8067	0.8192	**0.8259**	0.6914	**0.6920**	0.7408	**0.7433**
Boosting	0.6450	**0.6491**	0.7686	**0.7694**	0.7701	**0.7717**	0.8457	**0.8464**	0.6692	0.6692	0.7397	**0.7412**
SVM	0.4959	**0.5077**	0.7240	**0.7253**	0.7201	0.7155	0.7674	**0.7710**	0.6236	**0.6240**	0.6662	**0.6687**

Table 5. Comparison of system built using Lexical and Syntactic, Semantic and combined features with standard tf-idf and with best performing system

System	**Datasets**				
	Answer-Answer	**Headlines**	**Plagiarism**	**Post-editing**	**Question-Question**
Tf-idf	0.4426	0.6649	0.6636	0.8001	0.1120
Baseline	0.4113	0.5407	0.6960	0.8261	0.0384
UWB	0.6214	**0.8188**	0.8233	0.8208	0.7019
Lexical and Syntactic features with PEA	0.6188	0.7543	**0.8252**	0.8408	0.5906
Semantic feature with PEA	0.6372	0.7601	0.7510	0.8386	**0.7147**
Combined features with PEA	**0.6528**	0.7847	0.8067	**0.8464**	0.7068

9. Conclusion and future scope

In this paper, a novel feature Phrase Entity Alignment has proposed and evaluated on SemEval2016 test data set. It divides the sentences into a set of phrases and comparison is performed on the phrases. The most similar phrases are aligned and the similarity between the phrases are measured. The proposed feature is evaluated and compared with baseline system and top performing system presented in SemEval 2016 workshop. From the obtained results, it is observed that the proposed framework is performing well compared with other state-of-art models. It also identified that the performance of the proposed model is low for headlines dataset. The probable reason is that the headlines words are eye catchy words and the words presented in the headlines may not reflect the actual content presented in the article.

According to STS task definition the scores are assigned based on the equivalence of concepts and on the importance of the concepts that are missing or differing in the sentences. PEA identifies the concepts but does not address the importance of a concept and the relationship among the concepts that exists within a sentence which need to be addressed. To boost up the performance of the proposed system for headlines dataset, there is a need to explore a wider variety of sources for semantic features.

References

[1] A. Eneko, B. Carmen, C. Claire, C. Daniel, D. Mona, A. Aitor Gonzalez, G. Weiwei, G. Inigo Lopez, M. Montse, and M. Rada, “Semeval-2015 task 2: Semantic textual similarity, english,

spanish and pilot on interpretability", In: *Proc. of the 9th international workshop on semantic evaluation,* Denver, Colorado, pp.252–263, 2015.

[2] H. Christian, R. Robert, and P. Xose de la, "Exb themis: Extensive feature extraction from word alignments for semantic textual similarity", In: *Proc. of the 9th international workshop on semantic evaluation,* Denver, Colorado, pp.264-268, 2015.

[3] Md Arafat Sultan, B. Steven, and S. Tamara, "Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence", *Transactions of the Association for Computational Linguistics,* Vol.2, No.1, pp.219-230, 2014.

[4] C. Meadow, *Text Information Retrieval Systems*, Academic Press, Inc. Orlando, FL, USA 1992.

[5] D. Bar, B. Chris, G. Iryna, and Z. Torsten, "Computing semantic textual similarity by combining multiple content similarity measures", In: *Proc. First Joint Conf. on Lexical and Computational Semantics,* Montreal, Canada, pp.435–440, 2012.

[6] B. Sriram, F. Dave, D. Enginr, F. Hakan, and D. Murat, "Short text classification in twitter to improve information filtering", In: *Proc. SIGIR conf. on Research and development in information retrieval,* Geneva, Switzerland, pp.841–842, 2010.

[7] V. Hatzivassiloglou, J. Klavans, and E. Eskin, "Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning", In: *Proc. Joint SIGDAT Conf. on Empirical Methods in NLP and Very Large Corpora,* MD, USA, pp.203-212,1999

[8] P. W. Foltz, K. Walter, and T. K Landauer, "The measurement of textual coherence with latent semantic analysis", *Journal Discourse Processes,* Vol. 25, Issue 2, pp.285-307, 1998.

[9] C. Meadow, B. Boyce, and D. Kraft, *Text Information Retrieval Systems*, second ed, Academic Press, 2000.

[10] P. D. Turney, "Mining the web for synonyms: PMI-IR versus LSA on TOEFL", In: *Proc. of the 12th European Conf. on Machine Learning*, Freiburg, Germany, pp.491-502, 2001.

[11] Z. Wu. and M. Palmer, "Verb semantics and lexical selection", In: *Proc. of the Annual Meeting Association for Computational Linguistics,* Las Cruces, New Mexico, pp. 133-138, 1994.

[12] M. Rada and C. Courtney, and S. Carl, "Corpus-based and knowledge-based measures of text semantic similarity", In: *Proc. of the American Association for Artificial Intelligence,* Boston, Massachusetts, pp.775-780, 2006.

[13] Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea, and K.Crockett, "Sentence similarity based on semantic nets and corpus statistics", *IEEE Transactions on Knowledge and Data Engineering.* Vol.18, Issue 8, pp.1138–1150, 2006.

[14] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity", *ACM Transactions on Knowledge Discovery from Data,* Vol. 2, No. 2, Article 10, July 2008

[15] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis", In: *Proc. of the 20th International Joint Conf. on Artificial Intelligence,* Hyderabad, India, pp.1606–1611, 2007.

[16] G. Weiwei and D. Mona, "Weiwei:A simple unsupervised latent semantics based approach for sentence similarity", In: *Proc. First Joint Conf. on Lexical and Computational Semantics*, Montreal, Canada, pp.586–590, 2012.

[17] B. Sumit, S. Shrutiranjan, and K. Harish, "sranjans: Semantic Textual Similarity using Maximal Weighted Bipartite Graph Matching", In: *Proc. First Joint Conf. on Lexical and Computational Semantics,* Montreal, Canada, pp.579–585, 2012.

[18] H. Lushan, A.L. Kashyap, F. Tim, M. James, and W. Johnathan, "UMBC EBIQUITY-CORE: Semantic Textual Similarity Systems". In: *Proc. Second Joint Conf. on Lexical and Computational Semantics,* Georgia, USA, pp.44-52, 2013.

[19] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis", *Journal of the American Society for Information Science,* Vol. 41, Issue 6. pp:391–407, 1990.

[20] E. Marsi, H. Moen, L. Bungum, G. Sizov, B. Gambäck, and A. Lynum, "NTNU-CORE: Combining strong features for semantic similarity", In: *Proc. Second Joint Conf. on Lexical and Computational Semantics,* Georgia, USA, pp.66–73, 2013.

[21] M.A. Sultan, B. Steven, and S. Tamara, "DLS@CU:Sentence similarity from word alignment", In: *Proc. of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland, pp.241–246. 2014.

[22] A.L. Kashyap, H. Lushan, Y. Roberto, S. Jennifer, S. Taneeya, G. Sunil, and F. Tim, "Meerkat Mafia: Multilingual and Cross-Level

Semantic Textual Similarity Systems", In: *Proc. of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland, pp.416–423, 2014.

[23] A. Lynum, P. Pakray, B. Gamback, and S. Jimenez, "NTNU: Measuring Semantic Similarity with Sublexical Feature Representations and Soft Cardinality", In: *Proc. of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland, pp.448-453, 2014.

[24] M.A. Sultan, B. Steven, and S. Tamara, "DLS@CU: Sentence Similarity from Word Alignment and Semantic Vector Composition", In: *Proc. of the 9th International Workshop on Semantic Evaluation*, Denver, Colorado, pp.148-153, 2015.

[25] H. Lushan, M. Justin, C. Doreen, and T. Christopher, "Samsung: Align-and-Differentiate Approach to Semantic Textual Similarity", In: *Proc. of the 9th International Workshop on Semantic Evaluation,* Denver, Colorado, pp.172-177, 2015.

[26] R. Barbara, P. Katarzyna, C. Krystyna, W. Wojciech, and A. Piotr, "Samsung_Poland_NLP_Team: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity", In: *Proc. of SemEval-2016*, San Diego, California, pp.602-608, 2016.

[27] N. Afzal, Y. Wang, and H. Liu, "MayoNLP: Semantic Textual Similarity based on Lexical Semantic Net and Deep Learning Semantic Model", In: *Proc. of SemEval-2016,* San Diego, California, pp.674–679, 2016.

[28] H. Itoh, "RICOH: IR-based Semantic Textual Similarity Estimation", In: *Proc. of SemEval-2016*, San Diego, California, pp.691–695, 2016.

[29] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. "Learning Deep Structured Semantic Models for Web Search using Clickthrough Data", In: *Proc. ACM International Conf. on Information and Knowledge Management*, San Francisco, California, USA, pp.2333-2338, 2013.

[30] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval", In: *Proc. ACM International Conf. on Information and Knowledge Management*, Shanghai, China, pp.101-110, 2014.

[31] V. Vapnik, *The Nature of Statistical Learning Theory*. Second Edition, Springer, New York, 2001.

[32] L. Breiman, "Bagging. Technical Report No. 421". *Partially supported by NSF grant DMS-9212419*, Department of Statistics. September 1994.