Regression Based Approaches for Detecting and Measuring Textual Similarity

Sandip Sarkar[1(✉)], Partha Pakray[2(✉)], Dipankar Das[1(✉)], and Alexander Gelbukh[3(✉)]

[1] Jadavpur University, Kolkata, India
sandipsarkar.ju@gmail.com, dipankar.dipnil2005@gmail.com
[2] National Institute of Technology, Aizawl, Mizoram, India
parthapakray@gmail.com
[3] Instituto Politécnico Nacional, Mexico City, Mexico
gelbukh@gelbukh.com

Abstract. Finding Semantic similarity is an important component in various fields such as information retrieval, question-answering system, machine translation and text summarization. This paper describes two different approaches to find semantic similarity on SemEval 2016 dataset. First method is based on lexical analysis whereas second method is based on distributed semantic approach. Both approaches are trained using feed-forward neural network and layer-recurrent network to predict the similarity score.

1 Introduction

Semantic textual similarity (STS) measures the similarity between the two text sequences. Since 2013 SemEval workshop attracts researchers from many research groups. Like previous years, the main aim of STS task is to predict the semantic similarity of two sentences in the range 0 to 5 where 0 represents completely different sentences and 5 denotes completely similar sentences [4,5]. In this year Semeval test dataset consists of five different categories with different topics and different textual characteristics like text length or spelling errors: answer-answer, plagiarism, postediting, headlines, and question-question. In SemEval workshop the organizers provide test and training dataset of 2016 along with previous year dataset. Participants can use previous year dataset to train their systems. System quality is determined by calculating the Pearson correlation between the system output values and gold standard values. The system described in this paper explores an alternative approach based on five simple and robust textual similar features. Cosine similarity is used as first feature and second feature simply count the number of words common to the pair of sentences being assessed. The third feature calculates levenshtein ratio needed to transform one sentence into another. METEOR (machine translation metric) is also used to find the similarity score. Finally we are trying to predict the similar score using Gensim [2] toolkit where words and phrases are represented by word2vec [14] language model.

R. Prasath and A. Gelbukh (Eds.): MIKE 2016, LNAI 10089, pp. 144–152, 2017.
DOI: 10.1007/978-3-319-58130-9_14

2 Related Work

Different types of approach have been proposed to predict semantic similarity between sentences based on lexical matching and linguistic analysis [10,11]. For lexical analysis, researchers used edit distance, lexical overlap and largest common sub-string [12] features. Syntactic similarity is another method to find sentence similarity. For syntactic similarity, dependency parses or syntactic trees are used. Knowledge based similarity is mainly based on WordNet. The drawback of knowledge based system is that WordNet is not available for all languages.

On other hand distributional semantics is also used in the field of similarity task. The main idea of distributional semantic is that the meaning of words can be depending in their usage and the context they appear in. The improvement of system can be achieved by stemming, stopword removal, part-of-speech tagging.

3 Dataset

SemEval 2016 organizer provides five types of evaluation dataset in monolingual sub task (i.e. News, Headlines, Plagiarism, Postediting, Answer-Answer and Question-Question).[1] The similarity score of those sentences are calculated by multiple human annotators on a scale from 0 to 5. The details statistics about SemEval monolingual test dataset are described in the Table 1.

Table 1. Statistics of STS-2016 test data

Type	Sentence pair
Answer-Answer	1572
Headlines	1498
Plagiarism	1271
Postediting	3287
Question-Question	1555

4 System Description

Our experiment is divided into three stages. In the first stage, different types of pre-possessing technique are used. Next we calculated semantic similarity score using five types of features. Finally our system trained using two neural networks (i) multilayer feed forward network; and (ii) layered recurrent neural network. Same layered architecture used for both networks and the size of the hidden layer is 10. The details about the feature set are described in the next section. Figure 1 describes the overall architecture of our system.

[1] http://alt.qcri.org/semeval2016/task1/index.php?id=data-and-tools.

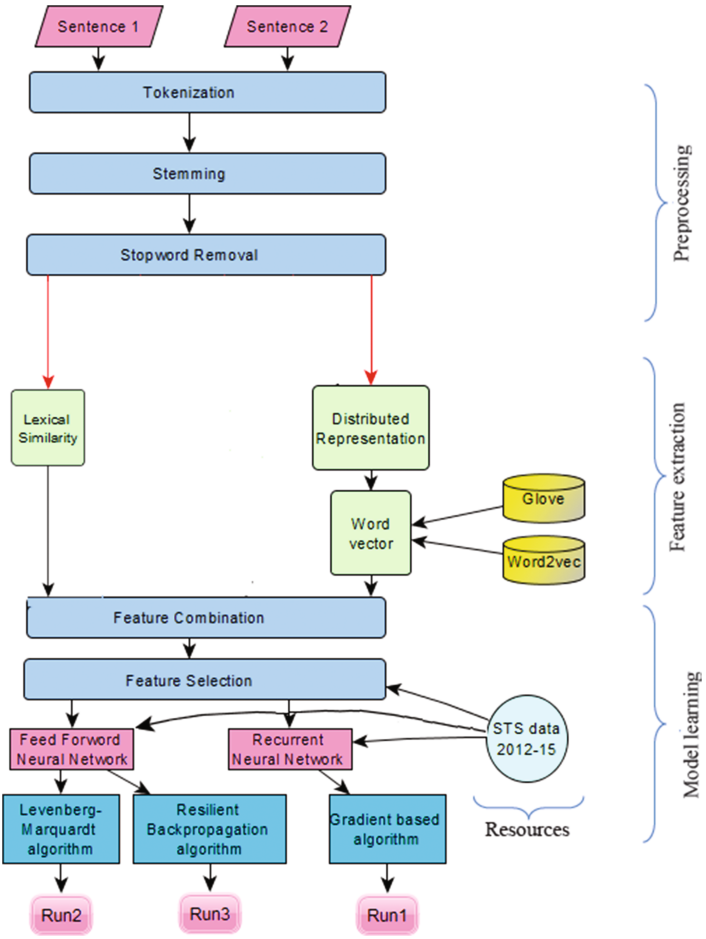

Fig. 1. System description

4.1 Preprocessing

In this section different types of pre-processing techniques are described like tokenization, stopword removal and stemming. The goal of this phase is to reduce inflectional forms of words to a common base form.

(a) **Tokenization**
Sentences can be divided into words only breaking at white-space and punctuation marks. But English language consists of many multi-component words like phrasal verbs. To solve this problem we used NLTK tokenizer. NLTK tokenizer is also required to remove stopword.

(b) **Stemming**
Stemming is an operation in which various forms of words are reduced to a common words. To improve the performance of Information Retrieval system steaming is also used.

(c) **Stop Words**
Stopwords are mainly common words in a language which contain less information. Words like 'a', and 'the' of are appears many times in documents. There is no universal list of stop words. We used NLTK stop word list for our system.[2]

4.2 Features

(a) **Cosine Similarity**
The most commonly used feature for the similarity score is the cosine similarity. In this approach each sentence is represented using vector space model. Cosine similarity is calculated using the dot product by the length of the two vectors. The details description about the cosine similarity is described in Table 2. The cosine similarity between two vectors (S_1, S_2) can be express using this mathematical formula:

$$S = \frac{S_1.S_2}{||S_1||.||S_2||} \tag{1}$$

Table 2. Cosine similarity

Sentence pair	Vector representation	Cosine similarity
Measure the depth of a body of water	Deep = 0, any = 0, measure = 1, the = 1, depth = 1, of = 2, a = 1, body = 1, water = 2, large = 0	0.51639
Any large deep body of water	Measure = 0, the = 0, depth = 0, any = 1, large = 1, deep = 1, body = 1, of = 1, water = 1	

(b) **Unigram matching ratio**
In this approach first total number of similar unigram between two sentences is calculated. Next the similar matching count is divided by the union of all tokens of those sentences. This feature is normalized because similarity score does not depend on the length of sentences. Table 3 describes about this feature where S1 and S2 denotes the sentence pair.

(c) **Levenshtein Ratio**
Levenshtein distance [8] is the difference between two strings. This distance is the minimum number of operation like insertions, deletions or substitutions needed to convert one string to another. Levenshtein distance is similar

[2] http://www.nltk.org/book/ch02.html.

Table 3. Unigram matching ratio

Sentence pair	(S1 ∩ S2)/(S1 ∪ S2)
Two green and white trains sitting on the tracks	.83333
Two green and white trains on tracks	

to Hamming distance but Hamming distance is only applicable to the similar length strings. The easiest way to calculate Levenshtein distance using dynamic programming. Levenshtein distance can be used in spell checking where a list of words can be suggest to the user whose levenshtein distance is minimal. The Levenshtein ratio of two strings a, b (of length $|a|$ and $|b|$ respectively) is expressed using Eq. 2. We use the Levenshtein ratio because Levenshtein distance is also depends on the length of the sentences. This feature describes in the Table 4.

$$\text{EditRatio}(a, b) = 1 - \frac{\text{EditDistance}(a, b)}{|a| + |b|} \tag{2}$$

Table 4. Levenshtein ratio

Sentence pair	Levenshtein distance	Levenshtein ratio
TSA drops effort to allow small knives on planes	6	.8958
TSA drops plan to allow small knives on planes		

(d) **Meteor**
Meteor automatic machine translation evaluation system release in the year 2004. Meteor calculates sentence level similarity by aligning them to reference translations and calculating sentence-level similarity scores. To improve the accuracy Meteor uses language specific resources like WordNet and Snowball steamers [6,7]. For our approach we used Meteor 1.5.[3] Meteor scoring is based on four types of matches (exact, stem, synonym and paraphrase).

(e) **Word2Vec**
In some region similarity between two sentences cannot be decided only using semantic and syntactic analysis. There is a semantic gap between the syntactic structure and the meaning of the sentences because of different vocabulary and language. So we need full knowledge and meaning representation. Using distributional semantic approach the gap between the syntactic meaning and original meaning can be removed. Recently researcher are using Gensim framework where words and phrases are represented using Word2vec [14]

[3] https://www.cs.cmu.edu/~alavie/METEOR/README.html.

language model. For our experiment we have used pre-trained word and phrase vectors which are available in Google News dataset [14]. The LSA word-vector mapping model contains 300 dimensional vectors for 3 million words and phrases. Gensim is a Python framework for vector space modeling. We have used Gensim for this experiment, and computed the cosine distance between vectors representing text chunks sentences from SemEval tasks.

5 Results

This section describes the results of our systems for English monolingual STS task of SemEval 2016. System performance measure using Pearson correlation. We used neural network to predict the STS scores. For training process all gold standard training and test data of the year 2012 have used in our task.

In Run 2 We trained our system using Levenberg-Marquardt algorithm and two layer feedforward network with 10 neurons in the hidden layer.[4] In Run 3 similar type of feedforward network is used but trained using Resilient Backpropagation algorithm [9].[5] Similarly in Run 1 our system trained using recurrent neural network [3].[6] However, this performance can be improved by increasing the training dataset and similar type of training and test dataset.

The detail result of the SemEval 2016 monolingual task using Word2vec feature is shown in the Table 5.

Table 5. System performance on SemEval STS-2016 monolingual data using Word2vec

	Run 1	Run 2	Run 3
Answer-Answer	**0.44468**	0.44258	0.44041
Headlines	0.55646	**0.57358**	0.54744
Plagiarism	0.78391	**0.79587**	0.77553
Postediting	0.77594	**0.78888**	0.75682
Question-Question	0.60747	0.61315	**0.62712**

Table 6 describes the result of cosine similarity feature on monolingual test dataset. The results also show that performance on monolingual dataset using only cosine similarity is not suitable for question-question test dataset.

Results in Table 7 show that our approach can achieve better performance except question-question dataset by combining different types of features (i.e. Unigram matching ratio, cosine similarity, lavenshtein ration and METEOR).

[4] http://nl.mathworks.com/help/nnet/ref/feedforwardnet.html.
[5] http://nl.mathworks.com/help/nnet/ref/trainrp.html.
[6] http://in.mathworks.com/help/nnet/ug/design-layer-recurrent-neural-networks.html.

Table 6. System performance on SemEval STS-2016 monolingual data using cosine

Corpus	Run 1	Run 2	Run 3
Answer-Answer	**0.42432**	0.41593	0.39188
Headlines	0.52655	**0.52840**	0.51711
Plagiarism	**0.68300**	0.66565	0.66364
Postediting	**0.80030**	0.78705	0.79928
Question-Question	0.13541	0.08708	**0.15116**

Table 7. System performance on SemEval STS-2016 monolingual data using Unigram matching ratio+METEOR+LR+cosine

	Run 1	Run 2	Run 3
Answer-Answer	0.52740	**0.56766**	0.52166
Headlines	0.69000	**0.71222**	0.66787
Plagiarism	0.73626	**0.77102**	0.72979
Postediting	0.75320	**0.77420**	0.74240
Question-Question	0.48372	**0.53835**	0.44357

On the other hand Table 5 shows that word2vec feature gives better result on question-question test dataset. With different type of feature set, we achieved a strong (>0.70%) correlation with human judgments on 3 of the 5 monolingual data set.

6 Compare with Winner Score and Baseline Score

Table 8 describes the comparison between the top ranked system and baseline score with our best result. In English Semantic Textual Similarity (STS) shared task the best result was obtained by Samsung Poland NLP Team.[7] Our System perform well for the postediting dataset. For postediting dataset the difference between the winner result and our result is minimum. However, our system struggles on both of the question-question and answer-answer dataset. Different combination of feature set gives better result on different type of dataset. When we are using word2vec then it gives better result for the question-question, postediting and plagiarism dataset. Similarly the score is high for answer-answer and headline dataset when cosine similarity, unigram matching ratio, levenshtein ratio and METEOR are used. Baseline system is based on unigram matching without stopword, METEOR and levenshtein ratio. In our approach cosine similarity and unigram mating ratio are added to baseline system.

[7] http://alt.qcri.org/semeval2016/task1/index.php?id=results.

Table 8. Compare with winner core and baseline score

Corpus	Winner score	Our system		Baseline score
		Best score	Features	
Answer-Answer	0.69235	0.56766	UMR+LR+METEOR	0.48023
Headlines	0.82749	0.71222	UMR+LR+METEOR	0.70749
Plagiarism	0.84138	0.79587	Word2vec	0.76752
Postediting	0.86690	0.80030	Cosine	0.77196
Question-Question	0.74705	0.62712	Word2vec	0.43751

7 Conclusions and Future Work

In this paper we described our experiment on the SemEval-2016 Task 1 monolingual test dataset in Textual Similarity and Question Answering Track. We observed that our system performance vary between different type of dataset. The Pearson correlation of all three runs are 0.8 or above for three test datasets: Headlines, Plagiarism, and Postediting, However the performance of our approach are comparatively lower for Question-question and Answer-answer test datasets. For the future work our aim is to analysis the reason behind the poor performance on answer-answer and question-question dataset. We also plan to include features which are directly based on Wordnet and also try to implement those features to find the similarity for crosslingual dataset.

Acknowledgment. This work presented here is under the Research Project Grant No. YSS/2015/000988 under Science and Engineering Research Board (SERB), Govt. of India. Authors are also acknowledges the Department of Computer Science & Engineering of National Institute of Technology Mizoram, India for providing infrastructural facilities and support.

References

1. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. Arch. Rat. Mech. Anal. **78**, 315–333 (1982)
2. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of LREC 2010 Workshop New Challenges for NLP Frameworks, p. 4550 (2010)
3. Elman, J.L.: Finding structure in time. Cogn. Sci. **14**(2), 179–211 (1990)
4. Agirre, E., Baneab, C., Cardiec, C., Cerd, D., Diabe, M., Gonzalez-Agirrea, A., Guof, W., Lopez-Gazpioa, I., Maritxalara, M., Mihalcea, R., Rigau, G., Uria, L., Wiebe, J.: SemEval- 2015 task 2: semantic textual similarity, English, Spanish and Pilot on interpretability. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 252–263 (2015)
5. Agirre, E., Baneab, C., Cer, D., Diab, M., Gonzalez-Agirree, A., Mihalceab, R., Wiebe, J.: SemEval-2016 task 1: Semantic textual similarity - monolingual and cross-lingual evaluation. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016) (2016)

6. Denkowski, M., Lavie, A.: Extending the METEOR machine translation evaluation metric to the phrase level. In: Proceedings of the HLT: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, pp. 250–253 (2010)
7. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, Ann Arbor, Michigan, pp. 65–72 (2005)
8. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Dokl. **10**, 707 (1996)
9. Riedmiller, M., Braun, H.: RPROP: a fast adaptive learning algorithm. In: Gelenbe, E. (ed.) International Symposium on Computer and Information Science VII, Antalya, Turkey, pp. 279–286 (1992)
10. Huang, A.: Similarity measures for text document clustering. In: Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC 2008), Christchurch, New Zealand, pp. 49–56 (2010)
11. Aziz, M., Rafi, M.: Sentence based semantic similarity measure for blog-posts digital content. In: 2010 6th International Conference on Multimedia Technology and Its Applications (IDC), pp. 69–74 (2010)
12. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures data warehousing and knowledge discovery. In: Proceedings of the 10th International Conference, DaWaK 2008, Turin, Italy, 2–5 September 2008, pp. 305–316 (2008)
13. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. **26**, 3111–3119 (2013)