FEATURE ANALYSIS FOR SEMANTIC TEXTUAL SIMILARITY

Dr. K. Anuradha[1], Himaja Indukuri[2], Tilak Putta[3]
Department of Computer Science and Engineering.
Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India

Abstract

The Semantic similarity is to measure the similarity between two texts. Semantic similarity is measured between two words, sentences, paragraphs and documents. In this paper, text chosen is sentences. Similarity measure is based on semantic and syntactic features. Semantics deals with the same meaning, syntactic deals with rules of syntax. In this paper, analyzed the impact of values syntactic and semantic features in measuring the Semantic Textual Similarity. The experimental work is carried out on SemEval 2017 datasets. The results show that in most of datasets a semantic feature, sent2vec has more impact.

Index Terms: Semantic textual similarity, syntactic, semantic, Bagging Model.

1. Introduction

Natural language processing is a way of communication between computer and human being through natural language.

In Natural Language Processing (NLP), semantic similarity plays a vital role and one of the fundamental tasks for many NLP applications and its related areas. The applications include machine translation (MT), Sentiment analysis, text summarization, question answering (QA), short answer grading, semantic search, Plagiarism, sentiment analysis.

Semantic textual similarity (STS) aims to estimate the semantic similarity between two sentences. Different syntactic, semantic, and structural similarity measures have been applied to estimate the similarity of texts. The similarity score ranges [0,5].0 indicates two sentences are completely independent, 1 indicates two sentences are not equivalent but are on same topic, 2 indicates two sentences are not equivalent but share some details,3 indicates two sentences are roughly equivalent but some important information differs, 4 indicates two sentences are mostly equivalent but some unimportant information differs and 5 indicates two sentences are completely equivalent [1]. Our system aims to quantify the similarity of pairs of sentences by encoding a variety of syntactic and semantic features in a vector of attributes and then predicting their similarity scores by employing machine-learning algorithms. The machine learning algorithm used in this paper is bagging regression approach discussed in section 3.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 describes bagging model. Section 4 describes syntactic and semantic features. Section 5 describes experiments. Section 6 describes the results for datasets with different features. Section 7 gives the conclusion of this work.

2. Literature survey

Vangapelli Sowmya et.al [5] has proposed a new syntactic feature, Phrase entity to improve semantic textual similarity**.** Pantulkar Sravanthi [6] proposed a feature based approach to find similarity between the sentence pair. Wael H. Gomaa [4] done survey of different approaches: string-based, Corpus-based and knowledge-based for measuring similarity between sentences. Ms.K.L. Sumathy et.al. [4] proposed two different frameworks, one for measuring document to document similarity and the other model which measures the similarity between documents to multiple documents. Aqeel Hussain [10] identified methods for textual similarity measurement. Three of these proposals are implemented. Two focuses on syntax similarity and one focus on semantic similarity. The two syntax algorithms represent edit distance and vector space model algorithms. The semantic algorithm is an ontology based algorithm, which lookup words in WordNet. Davide Buscaldi et.al [12] included some features based on alignment measures and tested different learning models, in particular Random Forests, which proved the best among those used in their participation.

3. Model

Bagging was proposed by Leo Breiman [2]. Bagging stands for **B**ootstrap **agg**regat**ing**. It is a machine learning ensemble method designed to improve accuracy by combining multiple predictors. It also reduces variance and avoids overfitting. Bagging works well for unstable learning rather than stable learning.

3.1 Steps for bagging algorithm

Step 1: Generates n new training data sets.

Step 2: Each new training data set picks a sample of observations with replacement (bootstrap sample) from the original data set.

Step 3: By sampling with replacement, some observations may be repeated in each new training data set.

Step 4: The n models are fitted using the above n bootstrap samples and combined by averaging the output for regression.

Sampling Example

Consider a data N= [15,18,26,9,34,54,12,42,]

Where N is the Original data with 8 elements. Now samples are taken from data with replacement.

Sample 1: [9,15,54,12,12]

Sample 2: [18,9,42,54,18]

Sample 3: [26,34,15,34,42]

Once samples are created model is built on each sample and average mean is the output of all bootstrap sample models.

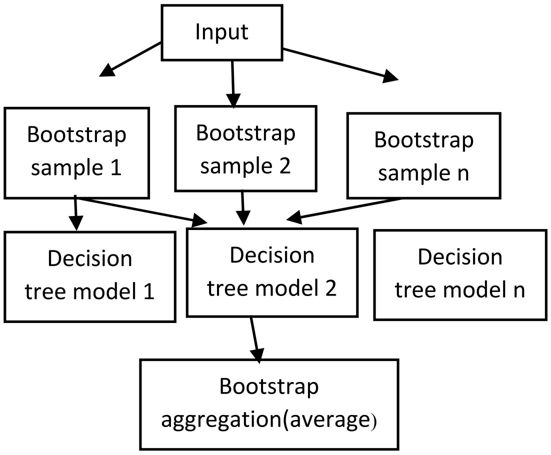

Fig 1: Flow of Bagging algorithm

4.Features

Features generated are syntactic and semantic features.Syntactic features are length features,Longest common subsequence(LCS), N-grams and TF-IDF .Semantic features are semantic with corpus,CDSSM (Convolutional pooling deep structured semantic model) and DSSM (Deep structured semantic model) using sen2vec and phrase entity.

4.1 Length features

For a pair of sentence A and B, |A|, |B|, |A-B|, |B-A|, |A-B|/|B|, |B-A|/|A|, |A∩B|, |AUB| are generated.

4.2 Longest common subsequence (LCS)

LCS of two sentences is the ratio between the maximal number of words that are shared in a sequence and the minimum length of the two sentences.

4.3 N-grams

N-gram is a contiguous sequence of *n* items from a given sequence of text or speech .An *n*-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" ; size 3 is a "trigram". Larger sizes are sometimes referred to by the value of *n* in modern language, e.g., "four-gram", "five-gram", and so on. Here 3-character Ngrams features,3 lemma N-grams and 3-word N-grams are generated. The N-grams are computed using Jaccard similarity as follows:

$$J(A,B) = \frac{|AUB|}{|A\cap B|} \tag{1}$$

4.4 TF-IDF

Tf-idf stands for term frequency-inverse document frequency

TF: Term Frequency, is the number of times the term occurred in document.

$$\text{TF(t)}=\frac{Number\ of\ \ times\ term\ t\ apperas\ in\ document}{Total\ number\ of\ terms\ in\ document} \quad (2)$$

IDF: Inverse Document Frequency, which measures how important a term in the document. While calculating TF all the terms in it are given equal importance. However, in IDF some terms give less importance such as ‘is’, ’and’, ’are’, ‘of’ because they appear lot of times. The IDF is calculated as follows:

$$\text{IDF(t)=loge}(\frac{Total\ number\ of\ documents}{number\ of\ times\ term\ t\ in\ doument}) \quad (3)$$

Finally, TF- IDF is computed as

TF-IDF(t)=TF*IDF. (4)

4.5 Phrase Entity: A phrase entity is formed by the combination of zero or one determiner, zero or more adjectives and a noun. Phrase entities are extracted from each sentence in the sentence pair. The semantic similarity is computed between each pair of sentence phrase entities using WordNet and brown corpus. The most semantically similar noun phrases are aligned [5].

4.6 Semantic with corpus

Semantic with corpus is important for calculating sentence similarity, because there is a need to weigh the importance of different words that occur in a sentence [8]. Different words contribute differently to the meaning of a sentence. This is especially important, because you need to keep all function words (for example “of”, “the”, “as” …), which contribute a lot less to the meaning of the sentence than other words. Words that occur more frequently in a corpus contains less information than words that occur less frequently [12]. The Semantic with corpus can be calculated as follows:

$$S=1-\frac{\log(n+1)}{\log(N+1)} \qquad (5)$$

Where n is the frequency of the word w in corpus and N is the total number of words in the corpus.

4.7 Sent2vec Features

Sent2vec generates Convolutional pooling deep structured semantic model (CDSSM) and Deep structured semantic model (DSSM) features. Both CDSSM and DSSM takes a bag of letter trigrams as input and maps the vector which are semantically similar. To find the similarity score Cosine similarity is computed as follows:

$$S = \frac{a1.a2}{||a1||*||a2||} \qquad (6)$$

Where a1 and a2 are the pair of sentences.

5.Experiment

5.1 Dataset

The datasets are taken from semeval 2017, provided 6 tracks of Monolingual and cross-lingual language pairs (Arabic-Arabic, Arabic-English, Spanish - Spanish, Spanish - English, English-English, Turkish-English). Table 1 depicts number of sentence pairs for each dataset. All the sentences are translated into English using Google Translator. This dataset is divided into 80% train data to build models and 20% test data to test the model. The data are divided using the random sample method.

Track	Language	Number of Pairs
Track 1 (Monolingual)	Arabic-Arabic (ar-ar)	250
Track 2 (cross lingual)	Arabic-English (ar-en)	250
Track 3 (Monolingual)	Spanish-Spanish (es-es)	250
Track 4a (cross lingual)	Spanish-English (es-en)	250
Track 4b (cross lingual)	Spanish-English (es-en)	250
Track 5 (Monolingual)	English-English (en-en)	250
Track 6 (cross lingual)	Turkish-English (tr-en)	500

Table 1: Monolingual and cross lingual datasets

5.2 Preprocessing

To generate features correctly data must be preprocessed. The preprocessing performs on both train and test data. The preprocessing steps include contractions replacement like don’t as do not, can’t as cannot, misspelling words, lowercase conversion.

5.3 Feature generation

For each sentence in the dataset syntactic and semantic features are generated. Out of 23 features 20 are syntactic and 3 are semantic features. Syntactic features include 3-character n-grams, 3-word n-grams, 3 lemma n-grams, lcs,8 length features and TF-IDF. Semantic features include semantic with corpus, sen2vec include CDSSM and DSSM, phrase entity.

5.4 Model building

The model is built on train data by combining all syntactic and semantic features. As output variable in a dataset is a continuous valued, regression algorithm is used. To build a model, data is preprocessed and then machine learning algorithm is applied on train data. The model is built using a bagging regression which is implemented in R.

5.5 Model evaluation

For model evaluation, the test pair sentences are preprocessed and then features are generated. These features are given as input to the model built. The model outputs the value ranges 0 and 5. Pearson coefficient correlation is used to evaluate accuracy. It is evaluated as follows:

$$r=\frac{N\sum XY-(\sum X)(\sum Y)}{[N\sum X^2-(\sum X)^2][N\sum Y^2-(\sum Y)^2]} \quad (7)$$

Where N represents the number of sentence pair, ∑X represents the sum of values in X and ∑Y represents the sum of values in Y, ∑XY represents sum of product scores of X and Y.

6.Results

Table 2 depicts the results of correlation of different features. For the datasets Arabic-Arabic, Arabic-English,4a English-Spanish and English-English, Sen2vec features has more impact and for Spanish- Spanish and 4b Spanish-English, Ngram features has the highest impact. For Turkish-English dataset TF-IDF has the highest impact. From the results, it is observed that sen2vec feature has more importance compared to other features.

Table 3 depicts the results of correlation between all syntactic, all semantic and all features. From the results it is observed that, for Arabic-Arabic, Arabic-English, Spanish-English, English-English, Semantic features has more impact compared to Syntactic features.

Features	Datasets						
	Track 1 (ar-ar)	Track 2 (ar-en)	Track 3 (es-es)	Track 4a (es-en)	Track 4b (es-en)	Track 5 (en-en)	Track 6 (tr-en)
Semantic with corpus	0.5969	0.5443	0.7273	0.7463	0.1141	0.6640	0.3519
Sen2vec	**0.6802**	**0.6560**	0.7786	**0.8178**	0.1055	**0.7793**	0.3159
Phrase entity	0.6595	0.5399	0.6402	0.5185	0.2910	0.6053	0.2112
Ngrams	0.6528	0.6101	**0.8452**	0.7174	**0.5556**	0.7477	0.2385
Length features	0.6071	0.5280	0.7636	0.6914	0.3133	0.7159	0.1506
Lcs	0.5752	0.4365	0.6946	0.6498	0.0804	0.5842	-0.0110
Tf-idf	0.6275	0.5564	0.81500	0.6382	0.2168	0.7156	**0.6275**

Table 2: Pearson coefficient correlation for datasets with different features.

Features	Datasets						
	Track 1 (ar-ar)	Track 2 (ar-en)	Track 3 (es-es)	Track 4a (es-en)	Track 4b (es-en)	Track 5 (en-en)	Track 6 (tr-en)
All syntactic	0.6950	0.6285	**0.8306**	0.7273	**0.4326**	0.7520	**0.6950**
All semantic	**0.7134**	**0.6631**	0.7850	**0.8263**	0.1822	**0.7798**	0.3592
All features	**0.7490**	**0.6698**	0.7940	**0.8279**	0.3913	**0.7962**	0.2385

Table 3: Pearson coefficient correlation for syntactic, semantic and all features.

7. Conclusion

In this paper, Semantic features and Syntactic features were discussed. Semantic features are semantic with corpus and Sen2vec. Syntactic features are Length features, Ngrams, Longest Common Subsequence, TF-IDF and Phrase Entity. These features evaluated on Semeval 2017 datasets on Monolingual and cross-lingual datasets. From the results, it is observed that semantic features have more importance. Among all Semantic features Sen2vec has more impact. The Pearson coefficient correlation varies based on sentences.

References:

[1] Eneko Agirre, Daniel Cer, Mona Diab, Bill Dolan, Aitor Gonzalez-Agirre," A pilot on Semantic Textual Similarity",http://www.cs.york.ac.uk/semeval-2016/task-6.

[2] Leo Brieman," Bagging predictors" Machine Learning, 24, 123-140 (1996).

[3] Wael H. Gomaa, Aly A. Fahmy," A Survey of Text Similarity Approaches", International Journal of Computer Applications (0975 – 8887) Volume 68– No.13, April 2013

[4] Ms. K. L. Sumathy ,Dr.Chidambaram," A Hybrid Approach for Measuring Semantic Similarity between Documents and its Application in Mining the Knowledge Repositories", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 8, 2016 .

[5] Vangapelli Sowmya, Bulusu Vishnu Vardhan, Mantena S.V.S. Bhadri Raju, Improving Semantic Textual Similarity with Phrase Entity Alignment, International Journal of Intelligent Engineering and Systems, Vol.10, No.4,2017

[6] Pantulkar Sravanthi, Dr.B.Srinivas,"semantic similarity between sentences", International Research Journal of Engineering and Technology (IRJET) ,Volume: 04 Issue: 01, Jan -2017 .

[7] Ming Che Lee, Jia Wei Chang, and Tung Cheng Hsieh," A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences", Scientific World Journal Volume 2014, Article ID 437162.

[8] 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.

[9] 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.

[10] Aqeel Hussain," Textual Similarity", Kongens Lyng by 2012 IMM-BSc-2012-16.

[11] 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.

[12]DavideBuscaldi,JorgeJ.Garc´ıaFlores,IvanV.MezaandIsaacRodr´ıguez,"Random Forest Regression for Semantic Textual Similarity task".

[13] P.Wiemer-Hastings, "Adding Syntactic Information to LSA," Proc. 22nd Ann. Conf. Cognitive Science Soc., pp.989-993, 2000.