Amrita_CEN at SemEval-2016 Task 1: Semantic Relation from Word Embeddings in Higher Dimension

Barathi Ganesh HB
Artificial Intelligence Practice
Tata Consultancy Services
Kochi - 682 042
Kerala, India
barathiganesh.hb@tcs.com

Anand Kumar M and **Soman KP**
Center for Computational Engineering
and Networking
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University, India
m_anandkumar@cb.amrita.edu
kp_soman@.amrita.edu

Abstract

Semantic Textual Similarity measures similarity between pair of texts, even though the similar context is projected using different words. This work attempted to incorporate the context space of the sentence from that sentence alone. It proposes combination of Word2Vec and Non-Negative Matrix Factorization to represent the sentence as context embedding vector in context space. Distance and correlation values between context embedding vector pairs used as a features for Support Vector Regression to built the domain independent similarity measuring model. The proposed model yielding performance 0.41 in terms of correlation.

1 Introduction

Semantic Textual Similarity (STS) assess the degree to which two snippets of text mean the same thing (Agirrea et al., 2015). The modules developed for successful STS systems have a broad range of potential applications including: Discourse Analysis, Information Retrieval, Machine Reading, Machine Translation, Question Answering, Text Summarization and Plagiarism Detection.

Degree of dependence between sentences vary even-though there exist similar words present in them (Example 1) whereas the dependence remains unchanged when the context is being projected with different words (Example 2). For instance,

S1 : Boy chases the cat.
S2 : Cat chases the boy.
Example 1

S1 : The rat jumps inside the tub.
S2 : Mouse dives into the vessel.
Example 2

From the above example, representing sentence as context dependent vector in a context space is more informative than the traditional frequency based representation methods. Thus by considering this, the proposed approach measures the similarity between the sentences as prescribed in STS task, which is given in the Table 1.

From the Table 1 it is clear that simple frequency based representation will fail to achieve the objective. Our approach proposes measuring similarity based on contextual information provided by the other words in the sentences instead of measuring similarity using just the words. Words tends to have different meaning with respect to their appearance with context.

In this proposed approach, sentence embedding will be found from word embedding in which words are represented as word embedding vectors with respect to context they occurs. Thereafter the similarity measure is done by finding correlation of the features in the sentence embedding.

Remaining paper details about the related works done on STS in section 2, detailed mathematical explanation is given in section 3 and statistics about the data-set, experiment and observations are explained in section 4.

2 Related Works

In this section we discuss the related works carried on the STS and how the current proposal has been built from the previous works.

Proceedings of SemEval-2016, pages 1290–1295,
San Diego, California, June 16-17, 2016.

Score	Similarity	Similarity Description	Sentence Pair
0	Different Topic	Different topics	S1: As long as it's not completely sealed air will get in S2 : Covers are also there to prevent things from getting in
1	Not Equal	Same topic	S1 : Actor Mickey Rooney dies aged 93 S2 : Ariel Sharon dies aged 85
2	Not Equal	Share some details	S1 : Putin opens Paralympics as protest staged S2 : Putin opens Winter Paralympics
3	Roughly Equal	Important information missing	S1 : India Ink: Image of the Day: July 2 S2 : India Ink: Image of the Day: March 4
4	Mostly Equal	Unimportant information missing	S1 : U.S. retailers agree to Bangladesh plant safety pact S2 : 70 retailers agree to new Bangladesh factory safety pact
5	Completely Equal	Means same thing	S1 : CIA chief visits Israel for Syria talks S2 : CIA chief in Israel to discuss situation in Syria

Table 1: STS Score Level for Similarity

The objective of the work is to represent the context of the sentence embedding from the word embedding of the sentence. To achieve this most of the recent research and also previous years works on paraphrase detection were based on deep learning or dimensionality reduction for semantic representation. This in turn was followed by a classification or regression to get the similarity score (Agirrea et al., 2015). Most of the distributional semantic representation based on dimensionality reduction algorithms (Han et al., 2013; Kashyap et al., 2015) and word embedding models were based on deep learning (Kenter and de Rijke, 2015; Wu et al., 2014; Socher et al., 2011).

The knowledge of WordNet and Latent Semantic Analysis (LSA) was integrated in-order to develop features for STS model SemEval 2013. This was done using distributional semantics (based on word's co-occurrence in different context) and semantic relation between the words in sentences (Han et al., 2013; Kashyap et al., 2015). Web corpus from the Stanford WebBase[1] project utilized to build the distributional semantic word representation and then the model was enhanced by integrating POS with WordNet. Same system was then extended to the Multilingual Semantic Textual Similarity and Cross Level Semantic Similarity in SemEval 2014 with few external resources (Google translate[2], Wordnik[3], and bing[4]) and showed greater accuracy (Kashyap et al., 2015).

[1]www-diglib.stanford.edu/ testbed/doc2/WebBase/.

[2]https://translate.google.co.in/.

[3]http://developer.wordnik.com/.

[4]https://www.bing.com/.

In order to represent the sentence pair, high quality word embedding was obtained using Word2Vec and Glove. Further feature vectors of length 60 computed using feature functions and evaluated on Microsoft Research Paraphrase Corpus (MSRP) (Kenter and de Rijke, 2015). As dealt with short text, similarity on long texts were found by computing non-linear semantic word representations on it. Thereafter it was fed to the Deep Semantic Embedding (DSE) to map long text into semantic space, where the semantic information was utilized to compute the similarity score (Wu et al., 2014).

Unfolding Recursive Auto encoder (U-RAE) along with dynamic pooling layer for fixed size representation was introduced for measuring the similarity between sentence pairs. Here it represents sentence as the parsed tree and words as word embedding. The pooled representation of sentences are then fed to the soft-max classifiers. The performance of this approach was evaluated using MSRP corpus and it attained the state of art accuracy (Socher et al., 2011).

In the above mentioned systems few tried to achieve the objective by using lexical information alone with high feature engineering, which seemed to have high manual effort and external resources. Other systems achieved greater accuracy by having context of the sentence either with the help of much external resources or with complex structure and computation. Finding mean or sum of the word embedding, are poor way to represent the context of the sentence. Our proposed approach simplifies the objective by relying only on word embedding and

matrix factorization. This approach is able to represent the context of the sentence with fixed size, which serves to be the essential and complex part of the objective.

3 Mathematical Representation

This section details how the vector representation of the word gives context information of individual lexicon in semantic space with respect to their co-occurring lexicons (3.1). It also shows the methodology for fixed size representation of the sentence embedding (3.2). It then deals with the feature functions (3.3) that is fed to the regression analyser(3.4).

3.1 Distributional to Distributed Representation

The phrase, "Distributional Semantics" means, representing a word with respect to the context that occurs across the corpus. Typically it was derived from dimensionality reduction algorithms (Singular Value Decomposition and other matrix factorization methods) applied on word - word context matrix (Turian et al., 2010). The represented vector in high dimension is sparse and dimension of the representation depends on the vocabulary of the word. This led to the research on word embedding representation.

Word embedding (Distributed Representation) is a low dimensional vector, which represents the word with respect to the context it occurs(Turian et al., 2010). Recent works have focused and shown proven results on distributed representation in-order to attain greater accuracy(Socher et al., 2011; Kenter and de Rijke, 2015). This is because, the model represents word as dense-low dimensional vector through non-linearity learning and negative sample learning for syntax - semantic information (Mikolov et al., 2013). Mathematically,

$$P(w_t|c) = softmax(score(w_t, c)) \quad (1)$$

$$= \frac{exp\{score(w_t, c)\}}{\sum_{w'} exp\{score(w', c)\}} \quad (2)$$

w_t is the vector representation for the word t in the vocabulary and $score(w_t, c)$ computes the like-mindedness of word w_t with the context c, where c represents the remaining words co-occurring with w_t. w_t is the word embedding with d dimension length, which is used to find the context embedding.

3.2 Word Embedding to Context Embedding

As discussed in previous section, the objective is to find the fixed size vector representation of the context from the sentence. Sentences may vary in length but their representation need to be in same length for further similarity measure. Here this is achieved by concatenating word embedding (context matrix or word embedding matrix) of the words in the sentence followed by the Non - Negative Matrix Factorization (NMF) (Lee and Seung, 1999). Given non-negative matrix V, NMF will factorize it into the basis matrix W and mixture matrix H, which is also a non-negative matrix. Mathematically,

$$V \approx WH^T \quad (3)$$

Where, V is $m \times n$ matrix, W is $m \times r$ basis matrix and H is $n \times r$ mixture matrix. Linear combination of basis vector (column vector) of W with weights of H gives the approximated context matrix (word embedding matrix) V. While factorizing, formerly random values are assigned to W and H then the optimization function is applied on it to compute appropriate W and H.

$$minf_r(W, H) \equiv \left\|V - WH^T\right\|_F^2 \quad (4)$$

$$s.t.\ W, H \geq 0$$

Where, r is the reduced dimension and F is the Frobenius norm. Here r fixed as 1 to have $d \times 1$ context embedding, where d is the dimension of the word embedding.

Each column vector in V is represented by a basis vector W weighted by the elements of H. This basis vector considered as context embedding vector, which is linearly combined with elements in the H to recompute the word embedding vectors with respect to its context. The non-negativity constraints makes interpretability straight forward than the other factorization methods. The basis vector in context space is not constrained to be a orthogonal, which is not affordable by finding singular vectors or eigen vectors. (Xu et al., 2003)

3.3 Feature Function and Decision Algorithm

Feature function measures the distance, dissimilarity and correlation between the context embedding pairs. Distance is measured to know, how close the

two context embeddings are in the context space, dissimilarity gives independence measure of context embeddings and correlation is carried over to know the dependency between context embeddings. Euclidean distance, BrayCurtis dissimilarity, City Block Distance, Chebyshev distance and Pearson correlation are considered in the feature function (Cha, 2007). For instance consider P and Q are the context embedding vectors of the two sentences and d is the dimension of the vector, then the measured functions given in the Table 2.

Measured Feature Functions
Euclidean Distance: $\sqrt{\sum_{i=1}^{d} \|P_i - Q_i\|^2}$
Bray Curtis Dissimilarity: $\frac{\sum_{i=1}^{d} \|P_i - Q_i\|}{\sum_{i=1}^{d} (P_i + Q_i)}$
City Block Distance: $\sum_{i=1}^{d} \|P_i - Q_i\|$
Chebyshev Distance: $\min_i \|P_i - Q_i\|$
Pearson Correlation: $\sum_{i=1}^{d} \frac{(P_i - Q_i)^2}{Q_i}$

Table 2: Measured Features

Attributes from the feature function is fed to the Support Vector Regression (SVR) to build the supervised similarity measure model. SVR is extended version of Support Vector Machine in-order to deal with regression problems (Welling, 2004). The advantages of the SVR here is, it doesn't make any assumption about data distribution, empirical risk minimization and has the ability to include non-linearity learning by changing the kernels.

4 Experiment

The Model diagram of the conducted experiment is given in Figure 1.

Statistics about the data-set are given in Table 3. Given data-set includes wide varieties of sentences in varying length and representation. This work is focused on building a unified model irrespective of domain. The training corpus for similarity measure involves shuffled sentence pairs from all the domains (i.e. single model for measuring similarity for Plagiarism, Answer-Answer, Post-editing, Headlines and Question-Question corpus).

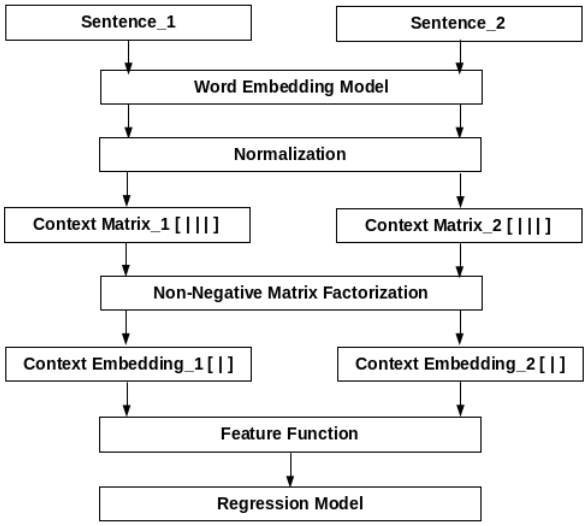

Figure 1: Model Diagram

To build the word embedding model, we use a snapshot of the articles in the English Wikipedia[5] (articles) has been utilized. After removing XML tags, special characters and unwanted spaces the corpus (size:12 GB) is fed to the Continuous Bag of Words (CBOW) model for processing (Mikolov et al., 2013). Window size, minimum occurrence and vector dimension are assigned as 5, 4, 400 respectively to create word embedding model (size:2.6 GB) using the Gensim package.[6]

The sentence pairs were fed to the word embedding model to represent the words in a sentence as vector of dimension 400. Word vectors in a sentence are concatenated to form a matrix (Context Matrix). Before concatenation the vectors are normalized (unity-based normalization) between 0 and 1, which forms dense positive vectors that are appropriate for further factorization. This is given by,

$$W' = \frac{W - min(W)}{max(W) - min(W)} \tag{5}$$

By equating the reduction rank to be one (r=1) the

[5]https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2 Downloaded on December 2015 (size:49.9 GB).

[6]https://radimrehurek.com/gensim/.

DataPerformance	# Training Sentences	# Test Sentences	Best	Median	Amrita_CEN
Plagiarism	1271	230	0.84138	0.78949	0.63336
Answer-Answer	1572	254	0.69235	0.48018	0.30309
Post-editing	3287	244	0.86690	0.81241	0.66465
Headlines	1498	249	0.82749	0.76439	0.43164
Question-Question	1555	209	0.74705	0.57140	-0.03174

Table 3: Data-set Statistics and System Performance in terms of correlation

NMF is carried out using the Nimfa package[7] on the context matrix to get the basis vector. This resultant basis vector of NMF is considered as the context embedding. This is because the linear combination of basis matrix along with mixture matrix will reconstruct context matrix. This can be visualized by generating words based on its context.

Once the context embedding pairs are found, they are fed to the feature function to measure the distance and correlation between them. These are used as attributes to train the SVR. SVR has been trained using Python Scikit-learn[8]. Radial Basis Function (RBF) kernel used for the non-linearity learning. The typical C = 1.0 and gamma = 1/length(training set) parameters are used in SVR.

While training, the performance of the system is measured by 10-cross validation. Correlation coefficient between gold-standard and predicted vector are computed to validate the significance of the system. The average correlation value obtained out of 10-cross validation during the training phase is 0.4178.

Our final system was trained on entire training corpus (9183 pairs) and then submitted to the STS shared task for evaluation. The official evaluation results are reported in Table 3. Our model performed poorly on the Question-Question data, but performed better on all the others. The model did best on the Plagiarism and Post-editing pairs. The average score of the system is 0.4090, which is almost equal to the training accuracy (0.4178).

5 Conclusion

A novel method for SemEval-2016 Monolingual Semantic Textual Similarity task has been described in this paper. Without depending on any resources or hand crafted features, it represents a simplified and unsupervised feature learning model for similarity measure. Our method performs well on the 2016 evaluation data except for the Question - Question corpus, however there is still room for improvement. The future work will be focused on more research on and the justification of context embedding derivation.

[7] http://nimfa.biolab.si/.

[8] http://scikit-learn.org/stable/.

References

Eneko Agirrea, Carmen Baneab, Claire Cardiec, Daniel Cer, Mona Diabe, Aitor Gonzalez-Agirrea, Weiwei Guof, Inigo Lopez-Gazpioa, Montse Maritxalara, Rada Mihalceab, et al. 2015. 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)*, pages 252–263.

Sung-Hyuk Cha. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc ebiquity-core: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 44–52.

Abhay Kashyap, Lushan Han, Roberto Yus, Jennifer Sleeman, Taneeya Satyapanich, Sunil Gandhi, and Tim Finin. 2015. Robust semantic text similarity using lsa, machine learning, and linguistic resources. *Language Resources and Evaluation*, pages 1–37.

Tom Kenter and Maarten de Rijke. 2015. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420. ACM.

Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic

pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Max Welling. 2004. Support vector regression. *Department of Computer Science, University of Toronto, Toronto (Kanada)*.

Hao Wu, Martin Renqiang Min, and Bing Bai. 2014. Deep semantic embedding. In *SMIR@ SIGIR*, pages 46–52.

Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM.