Sentence Encoders for Semantic Textual Similarity - A Survey

January 16, 2018

Abstract

Semantic similarity between two sentences is a basic language understanding problem that is applicable in many natural language processing applications. We replicated and extended existing works proposed in Association of Computer Linguistics' SemEval competitions to evaluate semantic models. We gathered features such as the length, vector spaces and text difference and implemented three models: SVM, RF, and CNN to measure semantic textual similarity.

1 Background

1.1 Semantic Textual Similarity (STS)

Semantic Textual Similarity (STS) is a task of finding how closely two sentences are related in terms of meaning Cer et al. (2017). Semantic similarity between two sentences is a basic natural language understanding problem that is applicable in many natural language processing applications such as web search, information retrieval, evaluation of machine translation system and automatic text summarization, etc,. Any natural language understanding problem starts with the challenge of describing words and sentences to machine understandable representation i.e a vectorial representations that encodes its whole meaning. These representations are also called distributional representations and the area of research on this problem is called Distributional Semantics. The model that learns these representation are called encoders. Since 1990, many vector space models have been proposed to estimate continuous representations of words such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA). Bengio et al. (2003) learnt the word representations while training neural language model. Later in 2008, the effective use of pre-trained word vectors in various task were explained by Collobert and Weston (2008). Mikolov et al. (2013) proposed a shallow neural network model for learning word representation which performed very well in capturing the general syntactical and semantic information. Recenlty, researchers have focused more in learning a general sentence representations that would capture its proper semantics.

To stream line the research and standardize the dataset related to semantic analysis task, Association of computer linguistics started organising shared task since 2012. For decade, traditional machine learning algorithms such as support vector machine or logistic regression were to used to solve the any NLP tasks.

For training and test of the STS model, we used data published in SemEval 2012-SemEval 2017 which includes the collection of STS dataset created using corpus from various domains like News headlines from the RSS feed of the European Media Monitor, Image captions from Flickr, Pairs of answers collected from Stack Exchange, student answers paired with correct reference answers from the BEETLE corpus, and Forum discussions about beliefs from the DEFT Committed Belief Annotation dataset **?**. The performance of the model is measured using Pearson correlation of the predicted score with the human judgment score given in the dataset.

1.2 Research Contributions

1.3 Contributions

2 Related Work

Despite of developing a number of learning algorithms for representing words and sentences, generating high quality and efficient sentence representations remains an unsolved problem until the present (Conneau et al., 2017). A efficient sentence representation that consist of the whole meaning with context is important as it can be used across various tasks with minimal adaption. This property results in smooth transfer of the learning to the model that learns any specific task. These sentence representations are used in many NLP task likes Semantic Textual Similarity, Paraphrase Detection, Sentiment Analysis, Sentence classification, as few are mentioned in here. The tasks are generalisation of various applications These I focus on STS task as Conneau et al. (2017) has proposed the natural language inference task appears to capture This work proposes a comparison study on different encoder models used for STS tasks and their importance in generating a

Since 2012, SEM-EVAL shared task for semantic textual similarity has been held by Association of Computer Linguistics to vitalize its research area. In 2012, supervised models based on the lexical and syntactic features of the sentence pair showed promising results on measuring semantic relatedness. These systems gave 52% - 59% correlation on various datasets by using regression models consisting of various similarity measure as its input features. Later, the unsupervised models did well for next two year in row using the WordNet knowledge and the LSA similarity measures which assume that the words with closer meaning highly co-occur in the text corpora. Han et al. (2013) proposed three approaches that involved LSA similarity model, semantic similarity model based on the alignments quality of the sentences and support vector regression model that had features from different combination of similarity measures and the measures from other two core models. It is observed that using n-gram overlap feature increased LSA similarity model. Out of three models proposed by Han et al. (2013), alignment based system gave 59%- 74.6% pearson correlation on four different dataset. Using this model's alignment quality as one of the feature in the Support Vector Regression model improved the correlation score to 78 %. Various supervised models using unigram/bigram overlap, vector distance, and cosine similarity of sentence embedding were proposed (Agirre et al., 2015).

Tian et al. (2017) proposed a system that adapted ensemble learning techiques to solve the Tectual Entailment and STS task using same set of features. The combination of classical NLP models like Support Vector Machine , Random Forest , Gradient Boost and a deep learning model are used in this system. For classical NLP models,single sentence and sentence pairs feature sets are hand engineered based on the properties like N-gram overlap, syntax, alignments, word sequence, word dependency, word representations, etc. In SEM-EVAL 2017, this mixed ensemble model gave 81 % pearson correlation outperforming all the neural models presented in that shared-task event.

Although using hand-crafted features with above mentioned models perform well, it has some drawbacks like tuning the features extracted on addressing the corpus from new domains, etc. Recent approaches in deep learning continues to prove the problem of semantic text matching can be handled in a efficient way Cer et al. (2017). The problem of distributional word match can be generalized to the problem of the distributional sentence match by using deep learning approaches. This helps in effectively learning the individual meanings from embedding of all the words in the sentence and derives a meaningful sentence representation from the word vectors. This section discusses about the top ranking models presented in SEM-EVAL 2017 that has been proposed to build sentence representations and predict sentence relatedness.

2.1 CNN Architecture

Shao (2017) presented a simple Convolutional neural network model for Semantic Textual Similarity task. This model constsis of CNN model and fully connected neural network (FCNN). CNN takes pre-trained word vectors from Glove Pennington et al. (2014) enhanced with handcrafted features as its input. It enhances word vector to task specific form in the convolutional layer and max-pooling generates the task-dependent sentence representation. FCNN generates the similarity score ranging from 0-5. This model ranked 3rd in SemEval-2017 with 78 % correlation on STS task.

2.2 Skip-ThoughtVector

Kiros et al. (2015) proposed Skip-Thought model based on skip-gram objective from Word2Vec Mikolov et al. (2013). For any three consecutive sentence in the document S_{i-1}, S_i, S_{i+1}, the Skip-Thought model predicts the previous sentence S_{i-1} and next sentence S_{i+1} given any sentence S_i. This work focus on training a encoder-decoder model. A varient of recurrent networks consisting of gated recurrent units (GRU) Cho et al. (2014) is used as a encoder to map input sentence into a generic sentence representations. RNN with conditioned GRU used as language model to decode the sentence representation and predict surrounding sentence S_{i-1} and S_{i+1}. On evaluating on semantic relatedness task, Skip-Thought outperformed all systems proposed in shared task SemEval 2014 Marelli et al. (2014) and was outperformed by dependency Tree-LSTM model.

2.3 LSTM Networks

Tai et al. (2015) proposed a recurrent neural networks(RNN) with tree based LSTM units. Child-Sum Tree-LSTM and N-ary Tree LSTM are two variants. Given a sentence synatctic structure in form dependency tree of the words, Tree-LSTM networks are capable of integrating the child node's information. The Tree-LSTM units in each node t consist of input gate i_t, output gate o_t, a cell unit c_t and a hidden output h_t. Unlike Standard LSTM, the parent node has one forget gate f_{tk} for each child node *k* in the Tree-LSTM. This property allows selective usage of child information. Previously proposed RNN models with sequential LSTM units have limited ability to capture meaning difference in the two sentence raised due to word order and synactical structures. Tree-LSTM address its issue by computing its hidden layer output as function of the outputs from its children hidden units and input vector.

On modelling semantic relatedness, the input x_t denotes the word vectors of sentence parse tree. The proposed model retains the information of more distant word from the current word compared to other exisiting models. These properties makes the model effective in highlighting the semantic heads in the sentence. It also captures the relatedness of two phrase which has no word overlap. With these propoerties, Tree LSTM performs better than existing sequential RNN-LSTM models and models with hand engineered features on predicting the semantic relatedness of two sentence.But one of the major downside is dependency tree-LSTM relies on parsers for dependecy tree input which computationally expensive to collect and does not exist for all languages making it inefficient in cross-lingual sentence representations.

2.4 Sent2Vec

Pagliardini et al. (2017) proposed a simple unsupervised objective Sent2Vec to train a generic distributed representations for sentences.The main contribution of Sent2Vec is its low computational cost for both training and inference relative to other existing state-of-art model. This model is a extension of CBOW training objective from Word2VecMikolov et al. (2013) to sentence context.

2.5 InferSent

Conneau et al. (2017) investigated on performance of various supervised encoders in learning universal sentence representations. They hypothesized that textual entailment task is good choice for learning the universal representations and demonstrated the hypothesize with various enoder models. To prove that sentence representations learnt are universal, the representations learnt from unsupervised and proposed hypothesis was used in 12 different transfer task namely Caption-Image retrievel, Paraphrase detection, Entailment/semantic relatedness, sentiment analysis, etc,. As the result of their experiments, Bi-LSTM with max-pooling trained on Natural Langugae Inference Task(Textual Entailment) using standford generated best sentence representations outperforming SkipThought Kiros et al. (2015) and FastSent Hill et al. (2016).

3 Proposed Work

A wide variety of supervised and unsupervised encoders for learning sentence representation are proposed by NLP researchers in recent times. However, there is a lack of knowledge about the encoding techniques that can capture useful, generic semantic information (Conneau et al., 2017). Supervised neural models captures the bias in the dataset effectively. This feature is a downside because it learns the task very well and forgets to capture general useful information over time leading to poor generalization. On the contrary, unsupervised learning models gives more importance to general information, therefore, failing to specialize the model for any specific task. Many reasons impact how the basic semantics of a sentence is being captured while learning. An important reason to note is the task for which the model is trained. Similarly, the encoder's architecture for both supervised and unsupervised neural models also impacts the learning in different ways. A comparison study on these encoder's architecture will enable us to gain better insight on how better sentence representations are captured.

In this project, I propose to perform a systematic comparsion of different encoder techniques for generating sentence representations and their abiltiy to capture semantics of the sentence. To do this, I am implementing and studying the following models: support vector machine (SVM), Random Forest (RF), Convolutional Neural Network Enoder Shao (2017) and BiLSTM RNN with max-pooling (Conneau et al., 2017). These models will be implemented using sci-kit learn, PyTorch and Keras library. As demonstrated in Conneau et al. (2017), Recognizing Textual Entailment (RTE) captures natural language inference, we will train our encoders on standford Natural Language Inference (SNLI) corpus Bowman et al. (2015) and SICK (Sentences Involving Compositional Knowledge) dataset Marelli et al. (2014) for semantic relatedness and RTE tasks.

The main objective of this comparison study is to understand the quality of the sentence representations and to answer the following questions:

- What are the vital features in prediction while using machine learning models ?
- Effectiveness of the traditional machine leanrning models ?
- What are the trade-offs that neural networks incur as opposed to the traditional machine leanrning models?
- What is the preferable neural network architecture for learning better sentence representations?
- Since the dimensionality is directly effect on the memory requirements and processing time, Which encoder ensures to learn good representations with what dimensionality size have good trade-off between accuracy and training time.
- What is the impact of various activations functions used in the enoders hidden layers?

4 Implementations

In this section, we will discuss the models that we are exploring. We implemented the models proposed in Shao (2017) and further extended them and measured their performance using Pearson's Correlation.

4.1 A Simple CNN Model For STS Task

This section explains the deep learning model used for semantic sentence similarity. The two main components of this model are convolution neural networks(CNN) based sentence representation model and fully connected neural networks(FCNN) used as the multi-class classifier. The CNN architecture consists of two convolution networks that work in parallel to mapping the two sentences to a vector space. The distributional vectors of the sentence pairs are used by FCNN to classify its sentence similarity score. In the following, we first describe our sentence model for mapping sentence pair to their intermediate representations and then explain how these representations are used to classify the relatedness score.

Sentence Model using CNN

Our CNN architecture for mapping sentences to feature vectors inspired from Shao (2017) is shown on Figure 1 . This architecture consists of two 1-dimensional convolution layer and a max pooling layer. The objective of this network is to convert the raw sentence into vector representations from **?**, pre-trained 300 dimension word embeddings of all the words $\{w_1, w_2, ..., w_{|s|}\}$ present in the sentence.

The input sentence to the convolution layers is treated as a sequence of real valued number where the real valued integers are retired from the integer-word mapping present in the vocabulary V. The vector representaion of all the word $w \in \mathbb{R}^d$ drawn from embedding matrix $W \in \mathbb{R}^{d\times|V|}$ in the embedding layer. To enhance the word representation with respect to this task, a true flag for word overlap is added as a additional dimension into the word vector representation for each word in the sentence. Then the CNN network applies convolution and max pooling operation to find the optimal feature vectos for the sentence that capture its semantics.

The idea behind the convolution layer is to learn the features which identifies the relationship between n-gram of the sentence using weight vectors *m* $\in \mathbb{R}^{|m|}$. The 1×1 weight vector *m* also known as filters of the convolution is used. This convolution operation is followed by applying Relu activation function to learn non-linear decision boundaries. This filters out the insignificant features learnt in previous operation. The output from convolution layer is passed to max pooling layer with pool size (1, $|S|$) where the semantic information learnt is aggregated and reduuces representation dimension from $1 \times |S| \times 300$(word vec dimension) to 1×300(word vec dimension).The convolution layers along with Relu

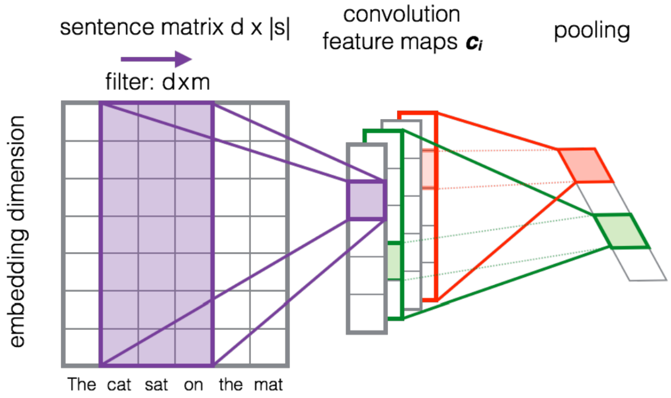

Figure 1: CNN Sentence Model Severyn and Moschitti (2015)

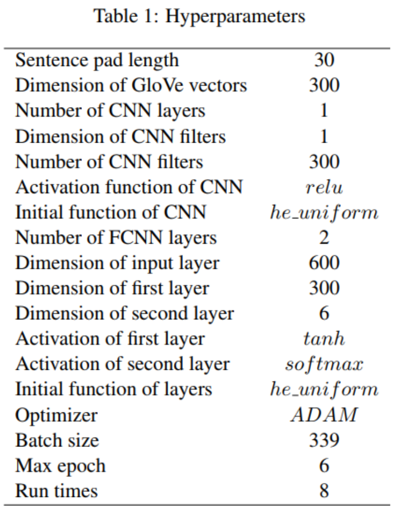

Table 1: Hyperparameters

Sentence pad length	30
Dimension of GloVe vectors	300
Number of CNN layers	1
Dimension of CNN filters	1
Number of CNN filters	300
Activation function of CNN	$relu$
Initial function of CNN	$he\_uniform$
Number of FCNN layers	2
Dimension of input layer	600
Dimension of first layer	300
Dimension of second layer	6
Activation of first layer	$tanh$
Activation of second layer	$softmax$
Initial function of layers	$he\_uniform$
Optimizer	$ADAM$
Batch size	339
Max epoch	6
Run times	8

Figure 2: Hyperparameters for FCNN Shao (2017)

activation function and max pooling acts as non linear feature detector for the given sentence. The output sentence representation from CNN is used to find Semantic Difference Matrix by performing a series of operation on the two sentence vector.

Semantic Difference Matrix

The semantic difference matrix is generated by concatenating the vector difference and vector product of two sentence representation. This matrix is used to classify the similarity measure using fully connected neural network(FCNN) with 2 dense layers.

$$SDV = (|SV_1 - SV_2|.(SV_1 \circ SV_2))$$

Similarity Measure using FCNN

This network consists of one hidden layer of size 300 nodes and a output layer of size 6.The hidden layer applies *tanh* activation function and the output layer applies softmax layer. The softmax layer calculates the probability over the six score labels. The hyper parameters of this network is shown Figure **??**

Finally, the model is trained using the categorical cross-entropy loss: given a vector of probabilities p for a training pair of sentences, if the correct similarity category corresponds to index i of p, the model will evaluate the loss as $L = log(pi)$.

5 Expected Results

6 Evaluation

7 Conclusion

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M Cer, Mona T Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *SemEval@ NAACL-HLT*, pages 252–263, 2015.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

Lushan Han, Abhay L Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc_ebiquity-core: Semantic textual similarity systems. In * *SEM@ NAACL-HLT*, pages 44–52, 2013.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*, 2016.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval@ COLING*, pages 1–8, 2014.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM, 2015.

Yang Shao. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133, 2017.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. Ecnu at semeval-2017 task 1: Leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197, 2017.