SURVEY ON SENTENCE SIMILARITY EVALUATION USING DEEP LEARNING

RAMAPRABHA J
SRM University, Kattankulathur

SAYAN DAS
Coviam Technologies, Bangalore

PRONOY MUKERJEE
SRM University, Kattankulathur

Abstract. Two questions asking the same thing can have different set of vocabulary set and syntactic structure. Which makes detecting the semantics equivalence between the sentences challenging. In online user forums like Quora, Stack Overflow, Stack Exchange, etc. its important to maintain high quality knowledge base by ensuring each unique question exists only once. Writers shouldn’t have to write the same answer to each of the similar question and the reader must get a single page of the question they are looking for. For example, consider questions like What are the best ways to lose weight?, How can a person reduce weight?, and What are effective weight loss plans? to be duplicate questions because they all have the same intent.

1. Introduction
Semantic Similarity is the degree by which linguistic terms are equivalent like document or sentences. Semantic Textual Similarity (STS) is the measure of similarity between documents or text. Application of semantic textual similarity can range from paraphrase identification to benchmarking machine translation. [1] For a computer to decide the semantic similarity between words, it should understand the semantics of the given words. Computer is a syntactic machine, which cannot understand the semantics. So it always made an attempt to represent the semantics words as syntactic words. Natural Language Processing, Artificial Intelligence, Machine Learning, cognitive science and psychology are the various fields along with academic community and as well as the production scale industries takes the advantage of Semantic Evaluation. [2] The amount of research on semantic similarity has increased greatly in the past 5 years, partially driven by the annual SemEval competitions (Jurgens 2014). SemEval (Semantic Evaluation) is International Workshop for Semantic Evaluation, organised under the umbrella of SIGLEX. Information resources provides the knowledge base for the computational task done by the various models or networks. Few of the most used information resources are listed below
1.1. Quoras Question Pair Dataset

To mitigate the inefficiency of redundant questions at the scale of millions, Quora needed an automated way of detecting these redundant question pairs. Dataset consists of over 400,000 lines of potential question duplicate pairs. Each line contains IDs for each question in the pair, the full text for each question, and a binary value that indicates whether the line truly contains a duplicate pair.

1.2. WordNets

WordNet provide the database links English nouns, verbs, adjectives, and adverbs to sets of synonyms that are in turn linked through semantic relations that determine word definitions. WordNet was created and being maintained at cognitive Science Laboratory of Princeton University.

1.3. Wikipedia

Wikipedia (http://en.wikipedia.org), is the largest encyclopaedia in existence. This can be mined for Explicit Semantic Analysis (ESA). This method uses machine learning to derive the high dimensional space of concept derived from Wikipedia. Over here the similarity between two words x and y is derived from the number of hits returned by the Google search engine for a given set of keywords. Keywords with similar meaning tends to be close in units of Normalised Google Distance. Specifically, the Normalised Google Distance (NGD) between two search terms x and y is:

$$NGD(x, y) = \frac{N}{f(x) + f(y)} - \frac{f(x, y)}{f(x) + f(y)}$$

where N is the total number of web pages searched by Google multiplied by the average number of singleton search terms occurring on pages. f(x) and f(y) are the number of hits for search terms x and y, respectively and f(x, y) is the number of web pages on which both x and y occur.
1.4. Vector Space Model
This method requires large corpora as the knowledge base and embed the knowledge in the form of Vector Space Models (VSM). A word embedding \( W: \text{words} \rightarrow \text{high-dimensional vectors} \) maps words in some language to high-dimensional vectors (generally 200 to 500 dimensions). For example, we might find:

\[
W(\text{cat}) = [0.1, 0.04, -0.92, ...]
\]

Various deep learning model architectures like Recurrent Neural Net Language Model (RNNLM), Feedforward Neural Net Language Model (NNLM), Continuous Bag-of-Words Model(CBOW), Continuous Skip-gram Model. For visualising these high dimension data, sophisticated techniques like t-SNE can be used to reduce the dimensionality of the vectors. In order to get compound features of embeddings, we first induce discrete clusters from the embeddings. Concretely, the k-means clustering algorithm is used. Each word is treated as a single sample. A cluster is represented as the mean of the embeddings of words assigned to it. Similarities between words and clusters are measured by Euclidean distance.

2. Proposed System

2.1. Support Vector Machines with ConvNets
Here the comparison are done by using traditional classifier like SVM with a CNNs. CNN are used to encode the word vectors in a such a way that similarity can be calculated by using a similarity metric. This particular model was used to study the semantic equivalence of 2 questions. The proposed CNN first transforms words into word embeddings , using a large collection of unlabelled data, and then applies a convolutional network to build distributed vector representations for pairs of questions. Finally, it scores the questions using a similarity metric. Pairs of questions with similarity above a threshold, defined based on a held-out set, are considered duplicates.

CNN is trained using positive and negative pairs of semantically equivalent questions. During training, CNN is induced to produce similar vector representations for questions that are semantically equivalent. here in this cosine similarity is used. The Textual Similarity Score is derived on the scale of 0-5, with 5 as most similar, thus making them paraphrases.

2.2. Siamese Recurrent Architectures
Siamese network consist of 2 identical networks each taking one of the two sentences. These identical network, also known as sister networks consist of encoder type RNN (Many to one type). Input layer for these RNN consist of word vector formed by tokenising the sentences, and searching each token over the embedding table. Given the input token \( W_t \) from any question statement. Now, \( X_t \) be the one-hot encoding for the given word. L be the embedding lookup matrix

\[
L \in \mathbb{R}^{V \times d}
\]

\( |V| \) is the vocabulary size. Then the vector corresponding to the token will be \( V_t = X_T \times L \) RNN based approach for language modelling provide state of the art performance, this mainly due to the capability to retain the context of sentences of arbitrary length. RNN maintain to retain the context of the sentences mainly due to the hidden states present at each time step. But training long chained RNNs may lead to minimising the gradient as we do back propagation, also known as vanishing gradient problem. [11]

This can be solved by using last cells instead of traditional cells with single conventional activation function. LSTM uses the concept of gates which works on the bounded context and only storing the information which is relevant [12].

The encoded sentences from the RNNs are then compared by some distance metric like manhattan distance, to throw some similarity value.
3. Challenges

- Same question/text can be paraphrased in multiple ways. [9]
- Two question/text can be asking for 2 different things but searching for same answer [9]
- CNN with SVM fails to identify the equivalence when the problem statement have image attached to it, which is striped out during the pre-processing [9]
- Vanishing Gradient Problem[11] for RNN language modelling. The error at a time step ideally can tell a previous time step from many steps away to change during backpropagation. In the case of language modelling or question answering words from time steps far away are not taken into consideration when training to predict the next word.
4. Conclusion
This paper represented a brief survey of various distance metric and deep learning models to evaluate the Semantic Textual Similarity[1]. We also explored the challenges faced on construction of these models. To conclude, in the field of computation linguistics, semantic evaluation has a broad set of applications.

References
[1] Agirre, Eneko, 2012. A Pilot on Semantic Textual Similarity. SemEval-2012, [Online]. Task 6, 385-393.
[2] Zhang, Shuang Zheng, Xuefeng Hu, Changjun. (2015). A survey of semantic similarity and its application to social network analysis. 2362-2367. 10.1109/BigData.2015.7364028.
[3] Maind, Ankush. (2012). Measurement of Semantic Similarity Between Words: A Survey. International Journal of Computer Science, Engineering and Information Technology. 2. 51-60. 10.5121/ijcseit.2012.2605.
[4] G. A. Miller, (1995) WordNet: A lexical database for english, Comm. ACM, Vol. 38, No. 11, 39-41.
[5] Gabrilovich, Evgeniy, 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. IJCAI, [Online]. 259, 1606-1611.
[6] R.L. Cilibrasi, P.M.B. Vitanyi, The Google Similarity Distance, IEEE Trans. Knowledge and Data Engineering, 19:3(2007), 370-383.
[7] Mikolov, Tomas Chen, Kai Corrado, G.s Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.
[8] Bogdanova, Dasha, 2015. Detecting Semantically Equivalent Questions in Online User Forums. Proceedings of the 19th Conference on Computational Language Learning, [Online]. 1013, 123131.
[9] Mueller, Jonas. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, [Online]. 1013, 1-7.
[10] Hochreiter, Sepp. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 6. 107-116.
[11] S. Hochreiter and J. Schmidhuber, LSTM can solve hard long time lag problems”, in Advances in Neural Information Processing Systems 9, ed. M. C. Mozer et al. (MIT Press, Cambridge MA, 1997), pages 473-479.