SURVEY FOR UNDERSTANDING SHORT TEXTS THROUGH SEMANTIC ENRICHMENT

Rutuja Subhash Gadekar[1], Prof. Bhagwan Kurhe[2]

[1] *M.E. Student, SPCOE, Otur, Pune*
[2] *Assistant Professor, SPCOE, Otur, Pune*

ABSTRACT

Short text messages such as tweets are very noisy and sparse in their use of vocabulary. Traditional textual representations, such as tf-idf, have difficulty grasping the semantic meaning of such texts, which is important in applications such as event detection, opinion mining, news recommendation, etc. We constructed a method based on semantic word embeddings and frequency information to arrive at low-dimensional representations for short texts designed to capture semantic similarity. For this purpose we designed a weight-based model and a learning procedure based on a novel median-based loss function. This paper discusses the details of our model and the optimization methods, together with the experimental results on both Wikipedia and Twitter data. We find that our method outperforms the baseline approaches in the experiments, and that it generalizes well on different word embeddings without retraining. Our method is therefore capable of retaining most of the semantic information in the text, and is applicable out-of-the-box.

Keyword: - *Short Text Similarity; Word Embeddings*

1. INTRODUCTION

Short pieces of texts reach us every day through the use of social media such as Twitter, newspaper headlines, and texting. Especially on social media, millions of such short texts are sent every day, and it quickly becomes a daunting task to find similar messages among them, which is at the core of applications such as event detection (De Boom et al. (2015b)), news recommendation (Jonnalagedda and Gauch (2013)), etc.

In this paper we address the issue of finding an effective vector representation for a very short text fragment. By effective we mean that the representation should grasp most of the semantic information in that fragment. For this we use semantic word embeddings to represent individual words, and we learn how to weigh every word in the text through the use of tf-idf (term frequency - inverse document frequency) information to arrive at an overall representation of the fragment.

These representations will be evaluated through a semantic similarity task. It is therefore important to point out that textual similarity can be achieved on different levels. At the most strict level, the similarity measure between two texts is often defined as being (near) paraphrases. In a more relaxed setting one is interested in topic- and subject-related texts. For example, if a sentence is about the release of a new Star Wars episode and another about Darth Vader, they will be dissimilar in the most strict sense, although they share the same underlying subject. In this paper we focus on the broader concept of topic-based semantic similarity, as this is often applicable in the already mentioned use cases of event detection and recommendation.

Our main contributions are threefold. First, we construct a technique to calculate effective text representations by weighing word embeddings, for both fixed- and variable-length texts. Second, we devise a novel median-based loss function to be used in the context of minibatch learning to mitigate the negative effect of outliers. Finally we create a dataset of semantically related and non-related pairs of text from both Wikipedia and Twitter, on which the proposed techniques are evaluated. We will show that our technique outperforms most of the baselines in a semantic similarity task.

We will also demonstrate that our technique is independent of the word embeddings being used, so that the technique is directly applicable and thus does not require additional model training when used in different contexts, in contrast to most state-of-the art techniques.

2. MOTIVATION

As [9] introduces, supposing that a practical and valid method of calculating the semantic difference between two short texts exists, there are many applications in Natural Language Processing (NLP) that can take advantage of it. For example, in the field of information retrieval and image retrieval from the Web, one of the best techniques for improving retrieval effectiveness is by using semantic similarity.

The use of text similarity is also useful for boosting accuracy results in relevance feedback and text categorization as for methods for automatic evaluation of machine translation, evaluation of text coherence [1], word sense disambiguation, formatted documents classification and text summarization. Also, it has been proved that for data sharing systems such as federated databases, message passing or data integration systems, web services, data management systems, etc., lexical and syntactical differences between shared variables can be solved by using semantic text similarity.

Semantic text similarity can also be used to build a text similarity join operator, that can be used to join two relations if their join attributes are textually similar to each other, which can be useful in several domains, such as integration of data from heterogeneous resources, mining of data, cleansing of data, etc. [3]

4. OBJECTIVES

The objective of this thesis is to determine and prove whether a system using word embeddings generated with GloVe can perform better than state-of-the-art systems that use the collection of models Word2Vec to build the word vector representations for their final use in the field of text similarity. We compare both methods (GloVe and Word2Vec) in several ways in order to determine which aspects of the word embeddings are different for the task of semantic text similarity. After analyzing the results, we also aim to use the currently generated word embeddings with GloVe in several different ways to improve the performance of our model.

5. RELATED WORK

In this section we discuss previous work related to the different aspects of our method.

Distributional semantics.

Distributional semantic approaches are based on the intuition that words appearing in similar contexts tend to have similar meanings. The Latent Semantic Analysis algorithm (LSA) [3] incorporates this intuition by building a word-document co-occurrence matrix and performing singular value decomposition (SVD) on it to get a lower-dimensional representation. Words are represented as vectors in this lower dimensional space. The distance between these word vectors (measured, e.g., with the cosine function) can be used as a proxy for semantic similarity. The full co-occurrence matrix, however, can become quite substantial for a large corpus, in which case the SVD becomes memory-intensive and computationally expensive.

Word vectors—also referred to as word embeddings—have recently seen a surge of interest as new ways of computing them efficiently have become available. In an algorithm called word2vec is proposed. There are two architectures to word2vec, continuous bag-of-words (CBOW) and Skip-gram. Both are a variation on a neural network language model [9, 2], but rather than predicting a word conditioned on its predecessor, as in a traditional bi-gram language model, a word is predicted from its surrounding words (CBOW) or multiple surrounding words are predicted from one input word (Skip-gram). To avoid computing a full softmax over the entire vocabulary, hierarchical softmax can be applied on a Huffman tree representation of the vocabulary, which saves calculations, at the potential loss of some accuracy. An additional strategy to get better embeddings is negative sampling, where, instead of only using the words observed next to one another in the training data as positive examples, random words are sampled from the corpus and presented to the network as negative examples.

An alternative way of getting word embeddings, called GloVe, is proposed. Rather than being based on language models it is based on global matrix factorisation. As such, it is closer to LSA, only a word-word co-occurrence matrix is used. GloVe avoids the large computational cost of, e.g., LSA by not building the full cooccurrence matrix, but training directly on the non-zero elements in it. As a cost function, the model uses a weighted least squares variant. The weighting function has two parameters, an exponent and a maximum cut-off value that influence the performance.

As both algorithms produce high-quality word embeddings and their implementations are publicly available, we use them in our experiments. Text-level semantics without external semantic knowledge.

Word embeddings, as described above, provide a way of comparing terms to one another semantically. It is not evident, however, how longer pieces of text should be represented with them. Several approaches have been proposed to go from word-level semantics to phrase-, sentence-, or even document-level semantics.

Le and Mikolov [8] propose a variation on the word2vec algorithm for calculating paragraph vectors, by adding an explicit paragraph feature to the input of the neural network. A convolutional neural network, built on top of word2vec word embeddings, is employed for modelling sentences in [6]. Other corpus-based methods have been proposed, such as [7], in which both semantic and string distance features are employed, and [10] in which a vector space model is used. All four methods, in line with the work presented here, do not rely on external sources of structured semantic knowledge, nor on natural language resources. As such, these methods are natural baselines for our experiments. It is problematic to reproduce the work presented in [8], however, as the original source code was not released by the authors and it is not clear, algorithmically, how the second step – the inference for new, unseen texts – should be carried out. Therefore, we omit this method as a baseline

Many methods rely on natural language resources such as parsers. Socher et al. propose recursive auto-encoders for the task of semantic textual similarity. This method relies on full parse trees for every sentence it processes. Annesi et al. [4] apply a kernel method on dependency parse tree features. Another strong method is presented in [2] where features from dependency parser are used to train a supervised method. The latter method, to our knowledge, yields the highest performance on the MSR Paraphrase Corpus [15, 1], an evaluation set commonly used for textual similarity experiments, and the one we use in our experiments in Section 5. Sentence representations based on word2vec word embeddings are also the focus in [9], where a convolutional neural network is trained on top of word2vec word embeddings. However, the method is only evaluated on sentence classification tasks (not on semantic similarity).

Text-level semantics with external knowledge.

A large body of research has been directed at using sources of structured semantic knowledge like Wikipedia and WordNet for semantic text similarity tasks. In [11, 12], methods very similar to one another are proposed, using pairings of words and Wordnetbased measures for semantic similarity. Our method of aligning words as described in Section 3 draws on this work. The key difference between these approaches and ours, apart from the fact that WordNet is used, is that parsing/POS tagging is carried out [12], as the WordNet-based measures are limited to comparing words having the same POS tag. Furthermore, no full-scale machine learning step is involved. All methods present one overall score, based on a threshold which is calculated through a simple regression step [12], or set manually [11].

Corpus methods are combined with WordNet-based measures in [13]. In [13] an IDF-weighted alignment approach, based both on WordNet-based and corpus-based similarities, is proposed. Texts are parsed and only similarities within identical part-of-speech categories are considered. Finally, a single score is calculated as an average over the maximum similarities. In a WordNet similarity measure is combined with word order scores. In neither approaches any machine learning step is applied.

SemEval STS.

Recently, the SemEval Semantic Text Similarity (STS) task [1] and SemEval STS task [2] were organised. A full description of the work of all participating teams (over 30 in both years) is beyond the scope of this section. We discuss the approaches of the best-scoring teams.

The best-scoring teams in both calculate a large number of features based on a wide variety of methods. Additionally, handcrafted rules are applied that deal with currency values, negation, compounds, number overlap and with literal matching [6]. The main difference with our approach, apart from the handcrafted rules, is in the features extracted, and in particular the number of additional resources required (WordNet, a dependency parser, NER tools, lemmatizer, POS tagger, stop word list, and WordNet, Wikipedia, Wiktionary, POS tagger, SMT system for three language pairs [6]).

In 2013, we see similar approaches where the best teams extract features from sentence pairs and use regression models (SVRs) to predict a similarity score. The features are based on LSA, WordNet and additional lists of related words and stopwords. In [8] features are calculated from aggregated similarity measures based on named entity recognition with WordNet and Levenshtein distance, higher order word co-occurrence similarity, the RelEx system, dependency trees and reused features of SemEval participants. Additionally, handcrafted features like lists of aliases (e.g., USA and United States) are used. A parallel between our work and both these approaches is the use of word alignment.

6. CONCLUSIONS AND FUTURE WORK

We have described a generic and flexible method for semantic matching of short texts, which leverages word embeddings of different dimensionality, obtained by different algorithms and from different sources. The method makes no use of external sources of structured semantic knowledge nor of linguistic tools, such as parsers. Instead it uses a word alignment method, and a saliencyweighted semantic graph, to go from word-level to text-level semantics. We compute features from the word alignment method and from the means of word embeddings, to train a final classifier that predicts a semantic similarity score

We demonstrate on a large publicly available evaluation set that our generic, semantics-only method of computing semantic similarity between short texts outperforms all baseline approaches working under the same conditions, and that it exceeds all approaches using external sources of structured semantic knowledge that have been evaluated in this dataset, to our knowledge

An important implication of our results is that distributional semantics has come to a level where it can be employed by itself in a generic approach for producing features that can be used to yield state-of-the-art performance on the short text similarity task, even if no manually tuned features are added that optimise for a specific test set or domain. Furthermore, the word embeddings, when employed as proposed above, substitute external semantic knowledge and make human "feature engineering" unnecessary. As our method does not depend on NLP tools, it can be applied to domains and languages for which these are sparse

It is interesting to see how other fields of research that deal with large corpora of unstructured text can benefit. For example, in automatically created probabilistic knowledge bases (e.g., [16]) triples are extracted from an input corpus and have a confidence score associated with them based on the number of sentences in the corpus describing the relation in the triple. Short text similarity can be used to improve this confidence score.

7. REFERENCES

[1] C. Quirk, C. Brockett, and W. B. Dolan. Monolingual machine translation for paraphrase generation. In EMNLP 2004, 2004.

[2] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In ICML 2008, 2008.

[3] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6):391–407, 1990.

[5] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain. Neural probabilistic language models. In Innovations in Machine Learning. Springer, 2006

[6] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In NIPS 2014, 2014

[7] A. Islam and D. Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. TKDD, 2008.

[8] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In Proceedings of the 13st International Conference on Machine Learning, 2014

[9] Y. Kim. Convolutional neural networks for sentence classification. In EMNLP 2014, 2014.

[10] P. Shrestha. Corpus-based methods for short text similarity. Rencontre des Étudiants Chercheurs en Informatique pour le Traitement automatique des Langues, 2011

[11] S. Fernando and M. Stevenson. A semantic similarity approach to paraphrase detection. CLUK 2008, 2008.

[12] R. Ferreira, R. D. Lins, F. Freitas, S. J. Simske, and M. Riss. A new sentence similarity assessment measure based on a three-layer sentence representation. In DocEng 2014, 2014

[13] R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In AAAI, 2006.