PROJECT PROPOSAL: SENTENCE EMBEDDING THROUGH DYNAMIC MIXING OF WORD EMBEDDINGS

Emad Sadeghi and Soroosh Khoram  
Department of Electrical and Computer Engineering  
University of Wisconsin - Madison  
{ssadeghi,khoram}@wisc.edu

Mohsen Mazrooyisebdani  
Department of Electrical and Computer Engineering  
University of Wisconsin - Madison  
mazrooyisebd@wisc.edu

1 INTRODUCTION AND EXISTING APPROACHES

There has been much research efforts in the recent years in the field of Natural Language Processing (NLP) to build on top of word embeddings, i.e., mapping words as vectors in some low dimensional space capturing lexical and semantic properties of words. Such embeddings can be obtained from the internal representations from neural network models of text or by low rank approximation of co-occurrence statistics (Arora et al., 2017). Much of these embeddings are built via unsupervised methods and are (almost) universal in the sense that they can be used in many different NLP tasks. Whether one can do embedding for other text structures like sentences, paragraphs, or even documents is still an unsolved problem (Arora et al., 2017).

The recent paper by Arora et al. (2017), shows a simple unsupervised approach to sentence embedding, based on the discourse vectors in the random walk model for generating text (Arora et al., 2016). The basic idea is to do the following: Use word embeddings computed using one of the popular methods on unlabeled corpus like Wikipedia, represent the sentence by a weighted average of the word vectors, and then modify a bit using PCA/SVD. It is noted in the paper that the suggested weighting improves the performance by about 10% to 30% in textual similarity tasks, and beats sophisticated supervised methods including RNN’s and LST’s (Arora et al., 2017).

2 OUR PROPOSED APPROACH

One possible limitation of the work by Arora et al. (2017) is that, when generating an embedding vector for a sentence, it uses a set of globally static weights for the words. A logical extension of this method, therefore, is to adopt a dynamic weighting scheme that learns mixing ratios for words based on the rest of the sentence, or other prior sentences. Formally, this dynamic scheme can be represented as:

\[ V_{S_j} = \sum_{i=1}^{k} \alpha_{w_i}(S_1, S_2, ..., S_k)V_{w_i} \]  

Here, \( w_i \) are the words in a dictionary of size \( k \), each of which having a corresponding \( V_{w_i} \) embedding vector. Using these vectors, we intend to learn the weights, \( \alpha_{w_i}(S_1, S_2, ..., S_m) \) (which are dependent on all of the sentences \( S_i \) in the dictionary that contain the word \( w_i \)), that best embed the sentence \( S_j \) with regards to a specific objective. One can imagine, for each sentence \( S_j \), a displacement with respect to \( \nabla S_j \) (defining the vector proposed by Arora), that is a (possibly) non-linear function of previous sentences. This function can then be learnt through an artificial neural architecture.

\[ \Delta V_{S_j} = g(\nabla S_j, S_1, S_2, ..., S_k) \]  

where \( V_{S_j} = \nabla S_j + \Delta V_{S_j} \).

Our goal in this study is to design such a neural architecture. In doing so, we will specifically focus on the sentiment analysis objective for this design and attempt to improve the results of the study.
by [Gan et al. (2016)]. We will further study the sentimental importance of each word on the overall sentiment of a sentence, essentially classifying them as “positive” or “negative”. Given the dynamic weights, this study involves identifying words with high average weights.

3 Timeline

In this section, we briefly describe our plan for completing this project. As a first step, we will aim on reading more about NLP to get ourselves acquainted with the vocabulary and tools/techniques used in this field, in addition to reading some of the requisite papers needed for understanding the paper by [Arora et al. (2017)] we aim to extend. In particular, we aim to learn about Recurrent Neural Networks (RNN) and learn how they are used for NLP tasks. To do this, we also need to learn how to implement such networks on Tensorflow and other relevant libraries used in the papers by [Arora et al. (2017), Lin et al. (2017), Wieting & Gimpel (2017), Gan et al. (2016), Arora et al. (2016), Arora et al. (2015), Pagliardini et al. (2017)], related to sentence embedding. Next, we will implement our proposed approach and compare our results with other methods.

REFERENCES


