Universal Sentence Encoder

Aditya Kumar

May 30, 2019

Word embeddings are now state of art for doing downstream NLP tasks such as text classification, sentiment analysis, sentence similarity etc and provides very good results compared to tf-idf or count vectorizer. Using word embeddings we can find the similarity between words and can apply vector operations and therefore can easily distinguish between {cat, dog, car}, i.e. cat and dog will be more similar compared to car.

But obtaining vectors for sentences is not immediate obvious. This article tries to explain approaches described in Universal Sentence Encoder. In this paper author describes two approaches to obtain sentence vectors.

1. **Deep averaging network (DAN):** Idea of DAN is described in this paper Deep Unordered Composition Rivals Syntactic Methods for Text Classification

   Word embeddings are low dimensional vector in N dimensional space which describe a word. To obtain vector space model for sentences or documents, appropriate composition function is required. Composition function is mathematical process of combining multiple words into single vector. Composition functions are of two types

   (a) Unordered: Treats as bag of word embeddings

   (b) Syntatic: Takes word order and sentence structure into account.

   Syntatic functions outperform unordered functions on many tasks but at same time it is compute expensive and requires more training time.

In this paper author introduces a deep unordered model that obtains near state of art accuracies on sentence and document level tasks with very less training time. It works in three steps:

(a) take the vector average of the embeddings associated with an input sequence of tokens

(b) pass that average through one or more feed-forward layer

(c) perform (linear) classification on the final layers representation

(d) Loss function is cross entropy.

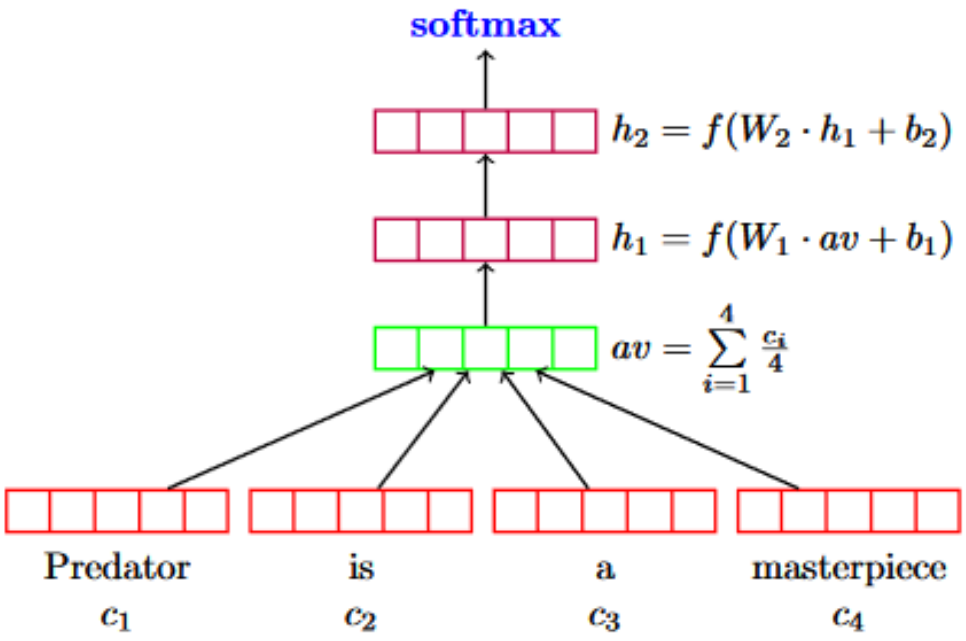

Figure 1: Deep averaging Network

Two important observations described in this paper are

- Accuracy can be improved by using a variant of dropout, which randomly drops some of words embeddings before averaging i.e. dropout inspired regularizer.

- The choice of composition function is not as important as initializing with pre-trained embeddings and using a deep network.

Here author tries to take best of both the approaches i.e. training speed of unordered function and accuracy of syntactic functions. DAN takes very less training time with slightly less accuracy on compared to other approach i.e. transformer encoder.

Observations on Results:

- Randomly dropping out 30% of words from the vector average is optimal for the quiz bowl task and results in 3% improved accuracy, which indicates that $p = 0.3$ is a good baseline to start with.
- DANs achieve comparable sentiment accuracies to syntactic functions and are trained in very lesser time compared to syntactic functions as RecNN
- 2-3 layers achieves good result for binary sentiment analysis task, but adding more depth is an improvement to shallow Neural bag of word model
- Sometimes it is very important to consider the ordering of words in NLP. "Man bites dog" and "Dog bites man" are two different sentences, but as we are just averaging the embeddings, those differentiations in sentences will be missed.
- Also DAN performed poorly on double negation sentences like "this movie was not bad". But at the same time DRecNN is slightly better in terms of polarity.

this movie was not good	negative	negative	negative
this movie was good	positive	positive	positive
this movie was bad	negative	negative	negative
the movie was not bad	negative	negative	positive

Figure 2: Negation

- On checking similarity of sentences "this is toy dog" and "this is dog toy", DAN encoding of both of these sentences should

be same as number of words are same and ordering should not matter, but it turns out that they are not same. Collab notebook code can be seen here.

This might be due to word dropout while averaging during

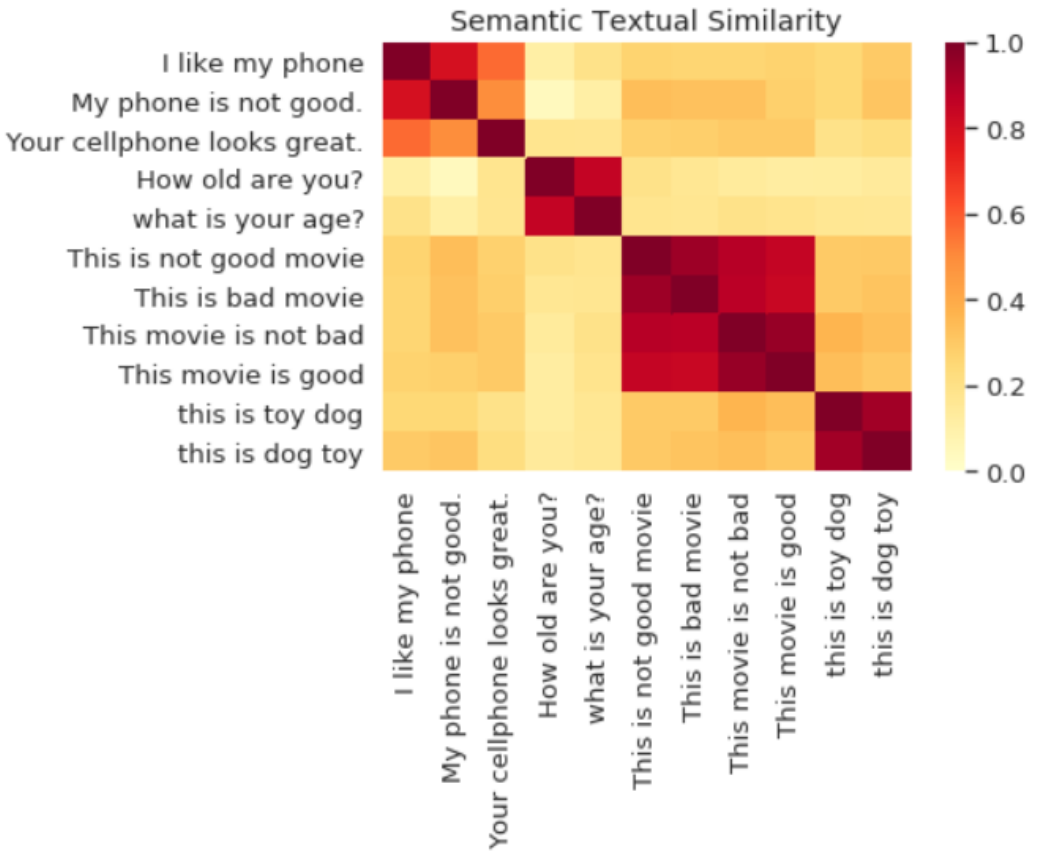

Figure 3: Textual similarity with DAN

feed forward pass of DAN.

References:

1. https://people.cs.umass.edu/ miyyer/pubs/2015_acl_dan.pdf