Report - Word Embeddings Project

1 Introduction

Word embeddings are based on the hypothesis that the context in which a word occurs provides sufficient information about the meaning of a word. They are used extensively in various NLP systems and have been shown to help improve performance on a number of tasks. These vector representations of words capture semantic and syntactic information by exploiting the distributional statistics of words. Words with similar meanings appear closer together in the vector space and vice versa. Mikolov et al. (2013) proposed the continuous bag of words model and the skip gram model which learn low dimensional vector representation of words. The bag of words model uses n words before and after the target word inorder to predict the word of interest while the skip gram model does the reverse: given a word, it tries to predict words within a window of a certain size surrounding the word. In trying to minimize its training objective, these models learn weights that can be used as low dimensional vector representations for individual words.

Bojanowski et al. (2016) develops on the skip gram model by making use of character n-grams. Instead of assigning a distinct vector to each word, each character n-gram is represented by a distinct vector and word vectors are obtained by by summing these vectors, thereby making use of the morphology of words.

In all of these methods the quality of embeddings is heavily correlated with the size of the training corpus and the number of distinct contexts in which a word occurs. This becomes a bottle neck when dealing with languages that lack extensive and quality training corpora. In this paper, we propose a method to overcome this problem by using a machine translation system to translate corpora from English to the language of choice. The translated data serves to augment the high quality original corpora that is available. Embeddings generated on the augmented corpus are then tested on a representative suite of extrinsic tasks (Nayak et al. 2016) such as part of speech tagging, named entity recognition, natural language inference, sentiment analysis and question classification and are shown to outperform facebook's pretrained fasttext embeddings on these eight Indian languages.We also make available, pretrained embeddings for eight Indian languages and hope to extend our work to all Indian languages.

2 Embedding Models

We compare the properties of word embeddings generated using the skipgram model of Mikolov et al. (2013) and the skipgram model of Bojanowski et al.(2016). In so doing, we restrict ourselves to standard settings that have been recommended by the authors.

2.1 Word2Vec

Mikolov et al.(2013) proposed two neural network based models for learning word embeddings, namely the continuous bag-of-words model (CBOW) and the skip-gram model. For predicting a particular target word w_t, the CBOW model uses a context window of n words on either side of the target word. On the other hand, given a word w_t, the skip-gram model predicts the n words surrounding

it. The skip-gram objective is thus to minimize:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n \leq j \leq n, \neq 0} \log p(w_{t+j}|w_t)$$

For the purpose of our experiments, we adhere to the skip-gram model as it is better than CBOW for infrequent words.

2.2 Fasttext

Bojanowski et al. (2016) builds on the skipgram model introduced by Mikolov et al.(2013). Words are represented as a bag of character n-grams. Special characters ¡,¿ are added at the beginning and end of each word to help distinguish n-grams from actual words. The model learns unique representations for each word and its character n-grams. The character n-grams can be used to obtain vector representations for words that were not encountered in the training data. For a given word w, the embedding is obtained using -

$$s(w, c) = \sum_{g \varepsilon G_w} z_g^T v_c$$

where G_w is a dictionary of n-grams corresponding to the word w and z_g is the vector representation of each of those n-grams.

3 Tasks

Embeddings obtained from the augmented corpus were compared using a suite of extrinsic tasks mentioned in Nayak et al. 2016. These include part-of-speech tagging, chunking, named entity recognition, sentiment classification, question classification and natural language inference. These tasks have been carefully chosen to evaluate the syntactic and semantic properties of the embeddings and to be representative of performance in relevant NLP tasks.

3.1 Sentiment Analysis

The Stanford Sentiment Treebank dataset was used for the sentiment analysis task. The model consists of an LSTM layer that is used to obtain sentence representations, followed by a fully connected layer and a softmax layer. The non-trainable embedding layer was initialised with the embeddings of interest and the results for various embeddings were compared.

3.2 Question Classification

The question classification task was performed as described in Li and Roth(2006). We only consider the coarse grained question classification task that involves 6 classes as opposed to 50 classes in the fine grained task. The model was trained on 4096 questions and evaluated on 542 questions and it consists of a basic LSTM followed by a single layered neural network and a softmax classifier.

3.3 Natural Language Inference

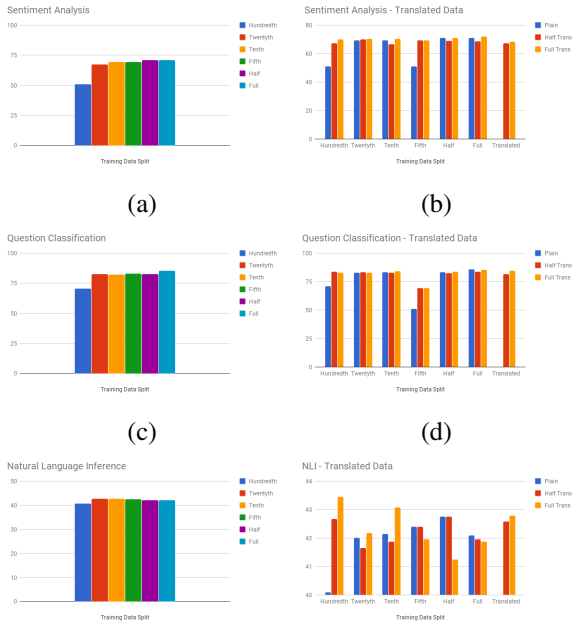
Given a natural language premise p, natural language inference refers to the task of reasonably inferring a natural language hypothesis h. To perform this task, sentence representations for the premise and hypothesis are obtained used two LSTM networks. These representations are then concatenated and passed through a fully connected layer followed by a softmax classifier.

4 Experimental Setup

We evaluated the embeddings on three different tasks - sentiment analysis, question classification and natural language inference. The embeddings were trained on different fractions of a wikipedia dump consisting 17005207 words, fractions of german to english machine translated europarl corpus consisting of 31653696 words as well as combinations of both.

5 Results

The graphs below show results obtained on the three tasks using our pretrained embeddings as initializations for the fixed embedding layer. In each of the tasks, the use of more natural data seems to directly correlated with performance. With embeddings trained purely on machine translated data, there is a small but noticeable improvement in performance. However, with embeddings trained on a combination of natural data and translated data, the improvement in performance is not always directly proportional to the size of the training corpus. The most noticeable improvements occur when we compare results obtained using just a hundredth of the natural data. It should also be noted that performances on all tasks do not improve proportionately. Tasks with lower baselines such as natural language



(a) (b)

(c) (d)

(e) (f)

Figure 1

inference are bound to show lower improvements in performance than sentiment analysis and question classification. Graphs (a),(c),(e) - refer to using different fractions of natural data alone. Graphs (b),(d),(f) represent the use of both natural and machine translated data.

Acknowledgments

References

Neha Nayak, Gabor Angeli, Christopher D. Manning. 2016. Evaluating Word Embeddings Using a Representative Suite of Practical Tasks

Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information Tomas Mikolov, Kai

Chen, Greg Corrado, Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space