Word2Vec using Character n-grams

Deepti Mahajan, Radhika Patil, Varsha Sankar
dmahaj@stanford.edu, radhikap@stanford.edu, svarsha@stanford.edu

CS 224n
Winter 2017

Introduction

The traditional Word2Vec model has lead to significant improvements in the field of NLP, but has a limitation that good vector representations are not learned for:

- Rare words
- Words that are not seen in the training corpus

This problem is more prominent in case of morphologically rich languages (e.g. German).

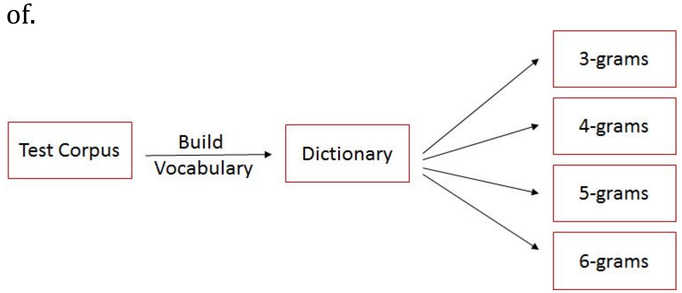
To overcome these, we incorporated the information about the character n-grams that each word is made of into its vector representation and assess the performance of the vectors using intrinsic and extrinsic evaluation methods.

Problem Statement

How could we improve Word2Vec representation for rare or unknown words without having to repeatedly train over large datasets?
We wanted to develop a way to derive reasonably good word vectors for new words from previously trained data.

Approach & Model

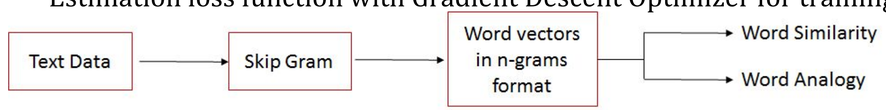
We trained embedding matrices of character n-grams (n =2 to 6) and for full words. The final word embeddings were obtained by adding the embedding for that word with the embeddings of all the ngrams that it is composed of.



We then use a modified skip-gram model to train the created word vectors. The probability of a context and center word occurring together was expressed in terms of the averaged n-gram vectors, as shown below. [3]

$$p(w_c \mid w_t) = \frac{e^{s(w_t,\, w_c)}}{\sum_{j=1}^{W} e^{s(w_t,\, j)}} \qquad s(w,c) = \frac{\sum_{g \in G_w} \vec{z}_g^{\,\top}}{|G_w|}\ \vec{v}_c$$

The model trained based on the derived word vectors, but updated the corresponding n-gram vectors. It used the Tensorflow Noise Contrastive Estimation loss function with Gradient Descent Optimizer for training.



Datasets

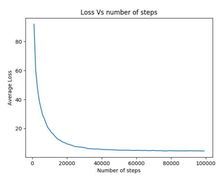
The 'text8' dataset was used for training. It consists of 100 MB (17 million tokens and ~254 thousand unique words) of cleaned English text taken from Wikipedia articles. Of these, a vocabulary of 50,000 was built. [1]

For word similarity evaluation, the WordSimilarity-353 Test Collection was used. The dataset consists of a set of 353 English word pairs. It also includes mean word similarity scores obtained from human subjects. [2]

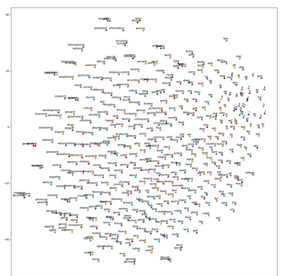
For word analogy evaluation, a dataset containing both semantic (e.g. France::Paris, Italy::Rome) and syntactic (e.g. small::smallest, large::largest) analogy pairs was used. The dataset was obtained from Google's word2vec code archives and contains 15,851 questions.

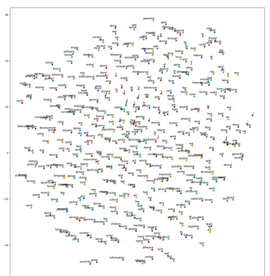
Experiments and Results

Training the model with a learning rate of 0.1 over 100000 steps produced the following learning curve



The embeddings were visualized in lower dimensions using TSNE - dimensionality reduction for word vectors obtained by normal word2vec and using char n-grams





We compared the cosine similarity of word pairs with the human-assigned scores using spearman correlation.

Example 3-gram word similarity groupings:
Nearest to american: americana, america, americans, americas, americo, erica, rican, african

Example skip-gram similarity groupings:
Nearest to american: dasyprocta, positions, encyclopedia, nonstandard, slain, fostered, city, arbor

The word analogy accuracy was computed based on 12,158 questions--those containing words in our 50,000 word vocabulary.

	word2vec (skip-gram)	3-gram model	2- to 6-gram model
Word Similarity	22.32	23.42	25.14
Word Analogy	34.2%	41.4%	43.1%

Future Work

In the future, the following could be implemented:
- Training the model for languages such as German, where many nouns are a combination of smaller nouns
- Evaluating the model using extrinsic NLP tasks, such as named-entity recognition
- Training the model on much larger datasets

REFERENCES:
[1] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin, "Placing Search in Context: The Concept Revisited", ACM Transactions on Information Systems, 20(1):116-131, January 2002.

[2] Mahoney, Matt. "About The Test Data". *Mattmahoney.net.* N.p., 2011. Web. 20 Mar. 2017.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606, 2016.

ACKNOWLEDGEMENTS:
We would like to thank Professor Manning, Dr. Socher, and our project mentor, Kevin Clark.