Name2Vec : Author Name Embeddings

Kalyan S. K. [1]

[1]DMKM, Universite Lumiere Lyon 2, Lyon, France

July 2, 2016

Abstract

Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing where words or phrases from the vocabulary are mapped to vectors of real numbers in a low-dimensional space relative to the vocabulary size ("continuous space"). Distributed representations of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks by grouping similar words [1]. My case study takes the work from our project and improves upon it by applying dimensionality reduction and clustering methods.

1 Introduction

Word2vec is a particularly computationally-efficient predictive model for learning word embedding from raw text. It comes in two flavors, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model.

Arithmetically, these models are similar, except that CBOW predicts target words (e.g. 'mat') from source context words ('the cat sits on the'), while the skip-gram does the inverse and predicts source context-words from the target words.

This inversion might seem like an arbitrary choice, but statistically it has the effect that CBOW smooths over a lot of the distributional information (by treating an entire context as one observation). For the most part, this turns out to be a useful thing for smaller datasets. However, skip-gram treats each context-target pair as a new observation, and this tends to do better when we have larger datasets. Since we have a smaller dataset we train a CBOW model.

2 Approach

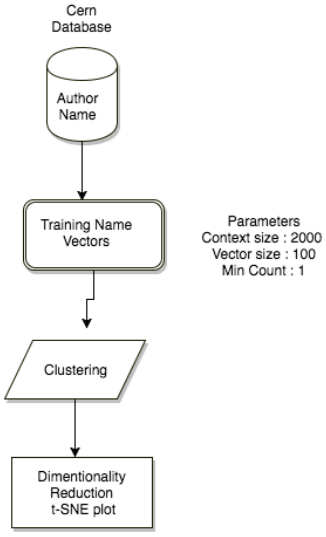

Figure 1: Workflow Description

Initially our dataset has around 2 million names. In this case study we have improved upon this by using more than 7 million author names. Maximum co-author for a paper was less than 50. Using Cern dataset we see that the maximum number of co-authors for a paper can easily reach upto 3500 since particle physicists tend to cite each other a lot and collaborate in large teams.

We also experimented with the compute time of the word2vec model and noticed a 3x improvement in speed for CBOW models compared to skip-gram models. However better representations are obtained when data large and hence skip-gram models are natural choice. Using this corpus we obtained a similarity matrix was obtained using word2vec [2]. This vector representation of author names were clustered using K-Means. We also applied dimensionality reduction techniques to represent author names in two dimensional plane. General solution work flow is described above in the figure.

3 Experiment

The word vector model trained for the case study has been trained on larger data compared to our project. The parameters used to train these name vectors are, min words as 1 as we have decided to keep all names even if they occur once. Context Size was chosen based on maximum co authors for an article. We set SG parameter as false as we want to train a CBOW model and vector length was chosen to be 100 (Default Value). We had a lot of memory issues on our local machines as glove could not handle this data volume. We were able to resolve this problem using a powerful server with 32GB ram and 4 cores.

Once we obtained similarity matrix of author name representation we applied K-Means clustering. Unfortunately the cluster labels obtained were unbalanced as 98 percent of the data fell under single cluster. We experimented with cluster centers of 3,5,10. We noticed the same imbalance in the cluster labels. Using 3 as the number of cluster center, we decided to shuffle the data and plot the first 200 observations. To plot this data we had to reduce the number of dimensions to 2. Using t-SNE plot we were able to achieve this, plot below shows vector representation of names colored by cluster association.

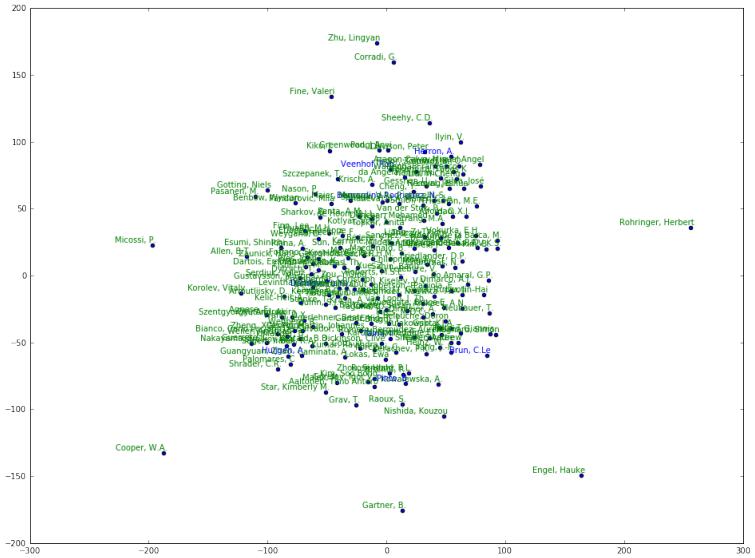

Figure 2: t-distributed stochastic neighbor embedding (t-SNE)

4 Future Work

Currently our Corpus size is very small. Using a larger data sets which contains more authors we could achieve better name vector representations. We could also try other clustering algorithms to check if they improve upon K-Means. Tuning the word2vec parameters might also help to improve the representations.

References

[1] Kai Chen Greg Corrado Jeffrey Dean Tomas Mikolov, Ilya Sutskever. Distributed representations of words and phrases and their compositionality. *KDD 2003*, 2013.

[2] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.