Mining Big Data - Intermediate Report

Investigating Paragraph Vectors

a1632538 Zachary Forman
a1646930 James Caddy

September 11, 2016

Introduction

This document describes our progress towards achieving our project goals. We have three primary project goals:

1. Replicate the results claimed by Le and Mikolov in their 2014 paper [1].
2. Compare `doc2vec` backed with k-nearest-neighbours with standard approaches to determining document similarity [2].
3. Apply `doc2vec` to source code, and explore how it distributes code segments.

Our work so far has mostly been towards the first and second of these goals.

Current Progress

Work on the IMDB dataset

Some exploratory analysis and testing, using gensim [3] as an implementation of `doc2vec`, and the tests have been run as per the paper [1]. So far the results are disappointing. Despite several attempts to replicate the paper's experiment, and obtain a working sentiment classifier, so far, we have not come close to achieving the accuracy described in the paper, or achieved by approaches. So far, only an accuracy of 0.68 has been obtained, in contrast with the 92% claimed by Le and Mikolov.

Work on the Wikipedia dataset

We again used gensim [3] to run `doc2vec` on the wikipedia corpus [4]. Even with only 2 training iterations on the corpus (compared with a desired 10-20), this took over 100 CPU hours. From the word and paragraph vectors trained on this data, we qualitatively evaluated the algorithm's performance on three axes:

1. Did the learned word vectors capture semantic meaning well?
2. Did the learned document vectors capture semantic meaning well?
3. Could the document vectors be used for similarity analysis?

Current Experimental Progress

All source code necessary to reproduce our results is available at `https://github.com/mbd-doc2vec-team/mbd-doc2vec/`, the IMDB dataset is available as referenced in Maas *et al.* 's work [5], and the wikipedia dataset is available at [4].

Experiments on the IMDB Dataset

The approach taken followed the method described in the paper as best as possible. Gensim [3] was used for the `doc2vec` implementation, with parameters as in Table **??** and Keras [6] backed with Theano was chosen for the neural network library, since neither were specified in the paper. It is unlikely that either of these choices were the cause for low performance, and more likely that our implementation was flawed. Both of these libraries were relatively easy to use which was good.
While the paper gives clear descriptions of most components and parameters in the classifier architecture, critical details are overlooked and explained poorly. Replicating the results without clear descriptions or values falls to trial and error, where any number of confounding variables could be in effect. One such ambiguity is the dimension of the document vectors, which can have large effects on accuracy.

Experiments on the Wikipedia Dataset

Future Work

IMDB Dataset

Further clarification of the architecture would improve the performance far better than continued tuning. Unless there is any insight disclosed in discussion around the paper, contacting the author of the paper seems like the only way this might be achieved. Which is unfortunate, given the author has been so far unwilling to help others in discussion groups. In order to obtain higher accuracy, there is the possibility that running the training for longer could help. Both the embedding and the classifier could benefit from this, however over-fitting the data is a concern. Currently, achieving an accuracy within 10% of that claimed by Le and Mikolov [1] seems unlikely, and as such, serves as a goal for our next experiments.

Wikipedia Dataset

Chromium Dataset

References

[1] Quoc V Le and Tomas Mikolov. “Distributed Representations of Sentences and Documents.” In: *ICML*. Vol. 14. 2014, pp. 1188–1196.

[2] Andrei Z Broder. “On the resemblance and containment of documents”. In: *Compression and Complexity of Sequences 1997. Proceedings*. IEEE. 1997, pp. 21–29.

[3] Radim Řehůřek and Petr Sojka. “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. `http://is.muni.cz/publication/884893/en`. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[4] Meta. *Data dump torrents — Meta, discussion about Wikimedia projects*. [Online; accessed 9-August-2016]. 2016. URL: `https://meta.wikimedia.org/w/index.php?title=Data_dump_torrents`.

[5] Andrew L. Maas et al. “Learning Word Vectors for Sentiment Analysis”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 142–150. URL: `http://www.aclweb.org/anthology/P11-1015`.

[6] *Keras: Deep Learning library for Theano and TensorFlow*. URL: `https://keras.io/`.