Mining Big Data - Data Collection Report

Investigating Paragraph Vectors

a1632538 Zachary Forman
a1646930 James Caddy

August 28, 2016

Data Sources

As we discussed in our proposal, we will use three primary data sources. Firstly, we use Maas' IMDB set [1] for verification and English language documents which can be found at `http://ai.stanford.edu/~amaas/data/sentiment/`. Secondly, we used The Wikimedia Foundation's English Wikipedia data dump [2] as a source for English language documents. We downloaded the full wiki dump from `https://dumps.wikimedia.org/enwiki/20160820/`. Lastly, we used The Chromium Project's Chromium repository [3] as a source for source code documents. We cloned the repository at revision 4fe31bb06cf458234d7017950a8b2b82427487c8.

Data Scale

The IMDB dataset contains a total of 100,000 files. Half are labelled as either a positive or negative reviews, and the remainder are unlabelled. It is 346MB in size and contains 23 million words, from a vocabulary of 90 thousand words.
The English Wikipedia Dataset is a single XML file, 13GB compressed and 55GB uncompressed. It contains 16 million pages, however this includes redirects and meta-pages such as categories and portals, and therefore only 5 million of these are useful as documents. Excluding markup, this dataset contains approximately 4 billion words.
The Chromium codebase is 2.6GB in size, and contains 245 thousand files. Of these, 23 thousand are C and C++ files, which contain 4.8 million lines of code and 608 thousand lines of comments.

Storage Format

The fundamental requirements for the `doc2vec` [4] algorithm to work are that each document can be associated with the words in the document, and that words in each document are just a series of words, i.e. no excess punctuation or markup.
The simplest way to meet these requirements is to store the data as a series of files, one per document, such that each file is named with the document's ID, and consists of a series of space separated tokens (words, in the case of English data, and code tokens in the case of code data). This advantage, while simple, scales well in the distributed case, with distributed filesystems like hdfs [5] providing an easy - and efficient - way to make stored data available over multiple nodes.

Data Processing

To transform the data into the format previously described, some data processing is required. All programs required to transform the datasets are available at `https://github.com/mbd-doc2vec-team/mbd-doc2vec/tree/master/data-processing`.
The IMDB dataset needed minimal processing. The bulk of the work had already been performed by the authors of the dataset. Additionally, punctuation, symbols and elements of markup were removed.
The Wikipedia dataset required extensive pre-processing, with each document needing to be parsed from the XML, and the plain text extracted from the wiki markup. Additionally, "fake" documents (e.g. portals, category pages) were removed.
The Chromium codebase required some pre-processing as well, with C++ files being extracted, the directory tree being flattened and the tokens extracted from each file.

Generating Similar Data

Given the plethora of sources of natural language data, data generation is not a high priority for us. Perhaps more importantly, simply duplicating given data is a known method for increasing the relevance of smaller documents, so to evaluate performance on larger datasets we can apply that technique. Despite these reservations, there are several promising ways to generate more data. Recurrent Neural Networks (RNNs) show great potential as a generative model (that differs from the word vector model used in `doc2vec`) of natural language and source code [6]. Similarly, Markov chains [7] provide a method of generating more data that retains many of the statistical properties of the original corpus.

References

[1] 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.

[2] 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.

[3] *Git repositories on chromium.* URL: https://chromium.googlesource.com/.

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

[5] Konstantin Shvachko et al. “The hadoop distributed file system”. In: *2010 IEEE 26th symposium on mass storage systems and technologies (MSST).* IEEE. 2010, pp. 1–10.

[6] *The Unreasonable Effectiveness of Recurrent Neural Networks.* URL: https://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[7] Claude Elwood Shannon. “A mathematical theory of communication”. In: *ACM SIGMOBILE Mobile Computing and Communications Review* 5.1 (2001), pp. 3–55.