THE JOHNS HOPKINS UNIVERSITY

human language technology
center of excellence

Jerboa: A Toolkit for Randomized and Streaming Algorithms

Benjamin Van Durme

TECHNICAL REPORT 7

MAY 28, 2012

©HLTCOE, 2012

Acknowledgements This work is supported, in part, by the Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

HLTCOE
810 Wyman Park Drive
Baltimore, Maryland 21211
http://hltcoe.jhu.edu

Jerboa: A Toolkit for Randomized and Streaming Algorithms

Benjamin Van Durme
Human Language Technology Center of Excellence
Johns Hopkins University
vandurme@cs.jhu.edu

Abstract

Recent studies have shown the applicability of streaming and randomized algorithms in a variety of large-scale language mining tasks. However, lack of many publicly available implementations of these methods has limited the use of these techniques beyond initial proof-of-concept, despite the growing interest in large-scale data. Jerboa is a Java-based toolkit aimed at providing reference implementations of a number of methods seen in recent literature, meant at enabling a greater adoption of these techniques for tasks in human language technology.

1 Introduction

The rise of big data has meant either a corresponding investment in computation, or for the less endowed researcher: a decision between either running fewer, longer experiments, or giving up on the benefit of massive datasets from lack of ability to exploit them. A recent thread in computational linguistics (CL) has been to develop or extend methods from the randomized and streaming algorithm communities, in order to add a "new knob" to existing NLP techniques: trading off the accuracy in underlying data representation in the name of memory savings and/or speed.

While randomized and streaming algorithms tend not to be any more complicated than recent efforts in, e.g., applications of non-parametric Bayesian inference, there is not nearly as large a diversity of software packages in the area. Jerboa is meant as an evolving toolkit providing reference implementations for a number of methods being reported on in the literature, to enable a wider use and exploration of large datasets through approximation. Being Java-based, it is aimed at ease-of-integration into existing NLP systems concerned with large-scale data mining.

Figure 1: Pygmy Jerboas (awaytrent, 2011)

The following provides first a note on the project name, then a brief survey of representative work.

2 Jerboa

Jerboas are a variety of small desert rodents found in North Africa and nearby parts of Europe and Asia. Movement is by hopping from their hind legs, described similarly to that of a kangaroo. The Pygmy Jerboa is referred to as the world's smallest rodent. The Jerboa toolkit represents a set of methods that often rely on random "hopping about" on a bit sequence, with the intent of being very small.

3 Filtering

Bloom filters (Bloom, 1970) are a well-known technique in the database community for hashing a keyset into a relatively small bit array, such that later queries result in purely one-sided error. Set-membership queries to a Bloom filter have guaranteed zero false negatives, while the rate of false positives increases in proportion to the number of unique keys to be stored in some fixed amount of memory.

Zaidan and Callison-Burch (2011) used Jerboa to store the entirety of the types in the Google n-gram collection (Brants and Franz, 2006) in a Bloom filter requiring roughly 10GB of memory. This allowed for a quick construction of an approximate boolean language model on phrases, reporting whether or a given phrase was observed in the Google collection. Recent work in specialized structures for large scale language modeling, such as KenLM (Heafield, 2011), have shown the ability to use similar amounts of memory while providing *counts* rather than boolean *presence*; the filtering application described here serves as an example of something that required 20 minutes to put together as part of standard data preprocessing, as compared to designing a data structure as a research project onto itself. There are a number of such utilitarian data management purposes that Jerboa can, and have, assisted with. For example, in a project ongoing, over 10 million webpages were rapidly filtered for duplicates, again requiring little time to put together a small tool on top of the underlying randomized methods in Jerboa.

4 Sampling

Reservoir sampling (first described in print by Vitter (1985)), is a method for collecting a sample of a fixed size k, taken uniformly at random from a sequence of prior unknown length. That is, when processing a stream of data $x_1, x_2, ...$, then with only $O(k)$ fixed memory and $O(1)$ time per element, at any point in the stream x_n the *reservoir* represents a uniform sample of all elements observed up to and including x_n. This is useful in large data-mining applications, as well as for manually exploring large experimental result collections: Jerboa includes a command-line routine that can be used on its own, or within pipes in conjunction with Unix tools such as `cat` or `grep`, in order to quickly pull a random set of lines from a file in a linear pass. As text data is often non-uniformly distributed within or across files (e.g., sorted alphabetically, by time, ...), then this allows for a quick, representative inspection often superior to the common habit of, e.g., reading the first 5 lines of output.

```
% grep "ing$" /usr/share/dict/words | head -5
aahing
abalienating
abandoning
abashing
abasing

% grep "ing$" /usr/share/dict/words | rand-sample 5
plate-bending
grouching
intershooting
exaggerating
coorieing
```

5 Counting

Morris (1978) gave a method for approximately counting large values with limited memory, for instance, using 8 bits to count over a range larger than 0...255 (see (Flajolet, 1985) for analysis). Jerboa contains an implementation of these Morris-style counters, along with the TOMB counter of Van Durme and Lall (2009a), which combined Morris-style counting with the Bloom filter count-storage method of Talbot and Osborne (2007).[1] Together this gives an online framework for tracking a very large keyset, such as all unique n-gram types up to some order, with accuracy a function of memory available. RandLM (Talbot and Osborne, 2007) is a C++-based language modeling toolkit that shares related functionality in this counting space.

TOMB counters led to an approach for *streaming PMI* introduced by Van Durme and Lall (2009b), then later explored by those such as Goyal and Daumé III (2011). Jerboa is a platform for replicating such prior work in order to properly evaluate novel data structures against established baselines.

Finally, Jerboa contains a Reservoir Counter (Van Durme and Lall, 2011) implementation, which competes against the Morris-style counting approach.

6 Similarity and Nearest Neighbors

Ravichandran et al. (2005) then Bhagat and Ravichandran (2008) introduced the use of Locality Sensitive Hashing (LSH) (Indyk and Motwani,

[1] Closely related constructions to TOMB counting were developed by Talbot (2009a) and Talbot (2009b).

1998) to the CL community for clustering distributionally related phrases. A challenge in that space is the large number of unique phrases, or inference rule paths, found in the language. This coupled with the high dimensional context vectors arising from large data sets. Van Durme and Lall (2010) then Van Durme and Lall (2011) gave methods for constructing such similarity *signatures* online, requiring significantly less memory than in prior work. These methods are available in Jerboa, along with routines for reading corpora or n-gram collections, in order to build up large signature sets for downstream use.

```
> the dog :: the cat
5513 3809       0.959
> the dog :: the tree
5513 4715       0.864
> visit toronto :: leave moscow
36 20           0.953
> visit toronto :: eat cheese
36 106          0.800
> light blue :: yellow
2694 29388      0.716
> the blue :: yellow
4944 29388      0.367
> new york city :: tokyo
9807 8213       0.845
> new york city :: rock
9807 38513      0.606
```

In the above, phrase pairs are queried against the top 50,000 million unique items (by frequency) in the n-gram collection of Lin et al. (2010), returning: (1) the number of distinct contextual features observed for each item, e.g., *the dog* was seen with 5,513 unique contexts; and (2) the approximate cosine score using a 512 bit LSH signature. Further, an implementation of the Point Location in Equal Balls (PLEB) owing to Indyk and Motwani (1998) is provided, allowing for fast approximate nearest neighbor queries.

Example applications based on this methodology include: streaming Topic Detection and Tracking (Petrovic et al., 2010), paraphrase acquisition (Chan et al., 2011), image/text hybrid dictionary acquisition (Bergsma and Van Durme, 2011), zero-resource spoken term detection (Jansen and Van Durme, 2011), and finally: large-scale monolingual distributional similarity computations (Ganitkevitch et al., 2012) based on the Annotated Gigaword collection (Napoles et al., 2012) and this toolkit.

7 Streaming Classification

As part of ongoing work in streaming analysis of social media, Jerboa includes an implementation of the Passive Aggressive online learning algorithms for regression (Crammer et al., 2003) and classification (Crammer et al., 2006). These are tied to an experimental pipeline for building *analytics* that analyze streams of data, such as emerging from Twitter users, while dynamically maintaining approximate, small-bit representations of classifier state (Van Durme, 2012).

8 Related Work

This toolkit is similar to that such as by Sekine (2008) and Lin et al. (2010), who gave frameworks for indexing and searching very large n-gram collections. Differing from those projects, Jerboa is built on a *streaming* model, rather than batch preprocessing, and where efficiencies in handling collections such as sets of n-grams come not from specialized data structures for that task, but from more general, *approximate* representations that apply to a variety of data sources and applications.

Beyond efforts previously mentioned, we close with pointers to further representative work in streaming and randomized algorithms for NLP. Streaming algorithms in Machine Translation were developed by Levenberg and Osborne (2009), Levenberg et al. (2010), and Levenberg et al. (2011). Further approaches to randomized language modeling were explored by Talbot and Talbot (2007) and Talbot and Brants (2008). Finally, Goyal et al. (2009), Goyal et al. (2010), and Goyal and Daumé III (2011) continue to explore various related approaches for a variety of NLP tasks.

9 Conclusion

Streaming and randomized algorithms are broadly applicable to large-scale processing of language, but there exist few supporting libraries that enable researchers to explore the impact of these methods in their own systems. Jerboa is a Java-based toolkit aimed at computational linguists, for more easily using these methods in their work.

Acknowledgments

Thanks to colleagues at Johns Hopkins University for providing regular developer feedback, and to Ashwin Lall with whom many of the underlying algorithms were developed.

References

awaytrent. 2011. Pygmy Jerboa from Egypt. Online video clip, YouTube. Uploaded February 12, 2011. Accessed on February 9, 2012. Creative Commons License.

Shane Bergsma and Benjamin Van Durme. 2011. Learning Bilingual Lexicons using the Visual Similarity of Labeled Web Images. In *Proceedings of IJCAI*.

Rahul Bhagat and Deepak Ravichandran. 2008. Large Scale Acquisition of Paraphrases for Learning Surface Patterns. In *Proceedings of ACL*.

Burton H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13:422–426.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1.

Tsz Ping Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking bilingually extracted paraphrases using monolingual distributional similarity. In *Proceedings of EMNLP: GEMS Workshop*.

Koby Crammer, Ofer Dekel, Shai Shalev-Shwartz, and Yoram Singer. 2003. Online passive-aggressive algorithms. In *Proceedings of NIPS*.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585.

Philippe Flajolet. 1985. Approximate counting: a detailed analysis. *BIT*, 25(1):113–134.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2012. Monolingual distributional similarity for text-to-text generation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*.

Amit Goyal and Hal Daumé III. 2011. Approximate scalable bounded space sketch for large data nlp. In *Proceedings of EMNLP*.

Amit Goyal, Hal Daumé III, and Suresh Venkatasubramanian. 2009. Streaming for large scale NLP: Language Modeling. In *Proceedings of NAACL*.

Amit Goyal, Jagadeesh Jagarlamudi, Hal Daumé III, and Suresh Venkatasubramanian. 2010. Sketch Techniques for Scaling Distributional Similarity to the Web. In *Proceedings of the ACL Workshop on GEometrical Models of Natural Language Semantics*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of EMNLP: Workshop on Statistical Machine Translation*.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of STOC*.

Aren Jansen and Benjamin Van Durme. 2011. Efficient spoken term discovery using randomized algorithms. In *Proceedings of ASRU*.

Abby Levenberg and Miles Osborne. 2009. Stream-based randomised language models for smt. In *Proceedings of EMNLP*.

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Proceedings of NAACL*.

Abby Levenberg, Miles Osborne, and David Matthews. 2011. Multiple-stream language models for statistical machine translation. In *Proceedings of EMNLP: Workshop on Statistical Machine Translation*.

Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. 2010. New Tools for Web-Scale N-grams. In *Proceedings of LREC*.

Robert Morris. 1978. Counting large numbers of events in small registers. *Communications of the ACM*, 21(10):840–842.

Courtney Napoles, Matt Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX 2012)*.

Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Proceedings of NAACL*.

Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. 2005. Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering. In *Proceedings of ACL*.

Satoshi Sekine. 2008. A linguistic knowledge discovery tool: Very large ngram database search with arbitrary wildcards. In *Proceedings of COLING*.

David Talbot and Thorsten Brants. 2008. Randomized language models via perfect hash functions. In *Proceedings of ACL*.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of ACL*.

David Talbot and John M. Talbot. 2007. Bloom maps. *CoRR*, abs/0710.3246.

David Talbot. 2009a. *Bloom Maps for Big Data*. Ph.D. thesis, University of Edinburgh.

David Talbot. 2009b. Succinct approximate counting of skewed data. In *Proceedings of IJCAI*.

Benjamin Van Durme and Ashwin Lall. 2009a. Probabilistic Counting with Randomized Storage. In *Proceedings of IJCAI*.

Benjamin Van Durme and Ashwin Lall. 2009b. Streaming Pointwise Mutual Information. In *Advances in Neural Information Processing Systems 22*.

Benjamin Van Durme and Ashwin Lall. 2010. Online generation of locality sensitive hash signatures. In *Proceedings of ACL*.

Benjamin Van Durme and Ashwin Lall. 2011. Efficient online locality sensitive hashing via reservoir counting. In *Proceedings of ACL*.

Benjamin Van Durme. 2012. Streaming Analysis of Discourse Participants. In *Proceedings of EMNLP*.

Jeffrey S. Vitter. 1985. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11:37–57, March.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from nonprofessionals. In *Proceedings of ACL*.