The Ngram Statistics Package (Text::NSP) - A Flexible Tool for Identifying Ngrams, Collocations, and Word Associations

Ted Pedersen[*]
Department of Computer Science
University of Minnesota
Duluth, MN 55812

Satanjeev Banerjee
Twitter, Inc.
795 Folsom Street
San Francisco, CA 94107

Bridget T. McInnes
College of Pharmacy
University of Minnesota
Minneapolis, MN 55455

Saiyam Kohli
SDL Language Weaver, Inc.
6060 Center Drive, Suite 150
Los Angeles, CA 90045

Mahesh Joshi
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Ying Liu
College of Pharmacy
University of Minnesota
Minneapolis, MN 55455

Abstract

The Ngram Statistics Package (Text::NSP) is freely available open-source software that identifies ngrams, collocations and word associations in text. It is implemented in Perl and takes advantage of regular expressions to provide very flexible tokenization and to allow for the identification of non-adjacent ngrams. It includes a wide range of measures of association that can be used to identify collocations.

1 Introduction

The identification of multiword expressions is a key problem in Natural Language Processing. Despite years of research, there is still no single best way to proceed. As such, the availability of flexible and easy to use toolkits remains important. Text::NSP is one such package, and includes programs for counting ngrams (count.pl, huge-count.pl), measuring the association between the words that make up an ngram (statistic.pl), and for measuring correlation between the rankings of ngrams created by different measures (rank.pl). It is also able to identify n-th order co-occurrences (kocos.pl) and pre–specified compound words in text (find-compounds.pl).

This paper briefly describes each component of NSP. Additional details can be found in (Banerjee and Pedersen, 2003) or in the software itself, which is freely available from CPAN [1] or Sourceforge [2].

2 count.pl

The program **count.pl** takes any number of plain text files or directories of such files and counts the total number of ngrams as well their marginal totals. It provides the ability to define what a token may be using regular expressions (via the `--token` option). An ngram is an ordered sequence of n tokens, and under this scheme tokens may be almost anything, including space separated strings, characters, etc. Also, ngrams may be made up of nonadjacent tokens due to the `--window` option that allows users to specify the number of tokens within which an ngram must occur.

Counting is done using hashes in Perl which are memory intensive. As a result, NSP also provides the **huge-count.pl** program and various other **huge-*.pl** utilities that carry out count.pl functionality using hard drive space rather than memory. This can scale to much larger amounts of text, although usually taking more time in the process.

By default count.pl treats ngrams as ordered sequences of tokens; *dog house* is distinct from *house dog*. However, it may be that order does not always matter, and a user may simply want to know if two words co-occur. In this case the **combig.pl** program adjusts counts from count.pl to reflect an unordered count, where *dog house* and *house dog* are considered the same. Finally, **find-compounds.pl** allows a user to specify a file of already known multiword expressions (like place names, idioms, etc.) and then identify all occurrences of those in a corpus before running count.pl

[*] Contact author : tpederse@d.umn.edu. Note that authors Banerjee, McInnes, Kohli and Joshi contributed to Text::NSP while they were at the University of Minnesota, Duluth.

[1] http://search.cpan.org/dist/Text-NSP/

[2] http://sourceforge.net/projects/ngram/

Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011), pages 131–133,
Portland, Oregon, USA, 23 June 2011. ©2011 Association for Computational Linguistics

3 statistic.pl

The core of NSP is a wide range of measures of association that can be used to identify interesting ngrams, particularly bigrams and trigrams. The measures are organized into families that share common characteristics (which are described in detail in the source code documentation). This allows for an object oriented implementation that promotes inheritance of common functionality among these measures. Note that all of the Mutual Information measures are supported for trigrams, and that the Log-likelihood ratio is supported for 4-grams. The measures in the package are shown grouped by family in Table 1, where the name by which the measure is known in NSP is in parentheses.

Table 1: Measures of Association in NSP

Mutual Information (MI)
(ll) Log-likelihood Ratio (Dunning, 1993)
(tmi) *true* MI (Church and Hanks, 1990)
(pmi) Pointwise MI (Church and Hanks, 1990)
(ps) Poisson-Stirling (Church, 2000)
Fisher's Exact Test (Pedersen et al., 1996)
(leftFisher) left tailed
(rightFisher) right tailed
(twotailed) two tailed
Chi-squared
(phi) Phi Coefficient (Church, 1991)
(tscore) T-score (Church et al., 1991)
(x2) Pearson's Chi-Squared (Dunning, 1993)
Dice
(dice) Dice Coefficient (Smadja, 1993)
(jaccard) Jaccard Measure
(odds) Odds Ratio (Blaheta and Johnson, 2001)

3.1 rank.pl

One natural experiment is to compare the output of statistic.pl for the same input using different measures of association. **rank.pl** takes as input the output from statistic.pl for two different measures, and computes Spearman's Rank Correlation Coefficient between them. In general, measures within the same family correlate more closely with each other than with measures from a different family. As an example *tmi* and *ll* as well as *dice* and *jaccard* differ by only constant terms and therefore produce identical rankings. It is often worthwhile to conduct exploratory studies with multiple measures, and the rank correlation can help recognize when two measures are very similar or different.

4 kocos.pl

In effect **kocos.pl** builds a word network by finding all the n-th order co-occurrences for a given literal or regular expression. This can be viewed somewhat recursively, where the 3-rd order co-occurrences of a given target word are all the tokens that occur with the 2-nd order co-occurrences, which are all the tokens that occur with the 1-st order (immediate) co-occurrences of the target. kocos.pl outputs chains of the form `king -> george -> washington`, where *washington* is a second order co-occurrence (of *king*) since both *king* and *washington* are first order co-occurrences of *george*. kocos.pl takes as input the output from count.pl, combig.pl, or statistic.pl.

5 API

In addition to command line support, Test::NSP offers an extensive API for Perl programmers. All of the measures described in Table 1 can be included in Perl programs as object–oriented method calls (Kohli, 2006), and it is also easy to add new measures or modify existing measures within a program.

6 Development History of Text::NSP

The Ngram Statistics Package was originally implemented by Satanjeev Banerjee in 2000-2002 (Banerjee and Pedersen, 2003). Amruta Purandare incorporated NSP into SenseClusters (Purandare and Pedersen, 2004) and added huge-count.pl, combig.pl and kocos.pl in 2002-2004. Bridget McInnes added the log-likelihood ratio for longer ngrams in 2003-2004 (McInnes, 2004). Saiyam Kohli rewrote the measures of association to use object-oriented methods in 2004-2006, and also added numerous new measures for bigrams and trigams (Kohli, 2006). Mahesh Joshi improved cross platform support and created an NSP wrapper for Gate in 2005-2006. Ying Liu wrote find-compounds.pl and rewrote huge-count.pl in 2010-2011.

References

S. Banerjee and T. Pedersen. 2003. The design, implementation, and use of the Ngram Statistics Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 370–381, Mexico City, February.

D. Blaheta and M. Johnson. 2001. Unsupervised learning of multi-word verbs. In *ACL/EACL Workshop on Collocations*, pages 54–60, Toulouse, France.

K. Church and P. Hanks. 1990. Word association norms, mutual information and lexicography. *Computational Linguistics*, pages 22–29.

K. Church, W. Gale, P. Hanks, and D. Hindle. 1991. Using statistics in lexical analysis. In U. Zernik, editor, *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*. Lawrence Erlbaum Associates, Hillsdale, NJ.

K. Church. 1991. Concordances for parallel text. In *Seventh Annual Conference of the UW Centre for New OED and Text Research*, Oxford, England.

K. Church. 2000. Empirical estimates of adaptation: The chance of two noriegas is closer to p/2 than p2. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 180–186, Saarbrücken, Germany.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

S. Kohli. 2006. Introducing an object oriented design to the ngram statistics package. Master's thesis, University of Minnesota, Duluth, July.

B. McInnes. 2004. Extending the log-likelihood ratio to improve collocation identification. Master's thesis, University of Minnesota, Duluth, December.

T. Pedersen, M. Kayaalp, and R. Bruce. 1996. Significant lexical relationships. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 455–460, Portland, OR, August.

A. Purandare and T. Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48, Boston, MA.

F. Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.