ACOPOST – A Collection of POS Taggers

1. BASIC INFORMATION

1.1 Tool name

ACOPOST.

1.2 Overview and purpose of the tool

ACOPOST is a free and open source collection of four part-of-speech taggers (`t3`, `met`, `tbt`, and `et`). In corpus linguistics, part-of-speech tagging (POS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up the words in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context — i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc. Once performed by hand, POS tagging is now done in the context of computational linguistics, using algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags.

For more details, see http://acopost.sourceforge.net/ and the user manual which is available in latex format at `acopost-1.8.4/doc/ug`.

1.3 A short description of the algorithm

Not applicable.

2. TECHNICAL INFORMATION

2.1 Software dependencies and system requirements

ACOPOST should work with LINUX.

2.2 Installation

A C compiler (gcc is recommended) is needed and the make program which are both most probably already installed on your machine if you're using UNIX. Some scripts use de Perl programming language which you want to have installed anyway.

Find a convenient place in your directory tree and unzip the archive which unpacks into a new directory acopost-x.y.z:

```
PROMPT> gunzip -c acopost-1.8.4.tar.gz | tar fxv -
acopost-1.8.4/
acopost-1.8.4/src/
acopost-1.8.4/src/Makefile
acopost-1.8.4/src/array.c
...
```

The fresh directory contains at least the following files and directories:

- Text file README with a short intro and latest changes.
- Directory bin which contains the Perl scripts and where the binaries are installed after compilation.
- Directory rc+ which contains the C files.
- Directory \verb+docs+ which contains the documentation, this user guide and a technical report \citep{Schroeder:2002b}.
- Directory \verb+examples+ which contains some example files.

To compile, change to the \verb+src+ directory and type \verb+make+. If everything works out ok, issue the command \verb+make install+ which installs the binaries into the directory \verb+../bin+. Congratulations! You're done.

If something goes wrong, try to fix it by adpating the \verb+Makefile+ or the source code. Don't forget to tell me about your problems so that I can provide a better solution with the next release.

You can now chose to add the \verb+bin+ directory as a full path to your \verb+PATH+ variable, to move/copy all binaries from the \verb+bin+ directory to a directory already in your \verb+PATH+ variable or simply decide to always use the full path to an \acopost program.

2.3 Execution instructions

For each of the four scripts, you have the purpose, the usage and features description. Please, see the user manual available in latex format at `acopost-1.8.4/doc/ug`.

2.4 Input/Output data formats

ACOPOST uses two file formats for text: raw and cooked.

- Raw text follows de line-based format described above but doesn’t contain any additional information:

  ```
  The rest went to inventors from France and Hong Kong .
  ```

- Cooked text contains the part-of-speech tags for the words. The tag immediately follows the word and the two are separated by one or more white space characters:

  ```
  The DT rest NN went VBD to T0 inventors NNS from IN France NNP
  and CC Hong NNP Kong NNP . .
  ```

2.5 Integration with external tools

Not applicable.

3. CONTENT INFORMATION

3.1 A test input file

The input file is in raw text.

3.2 The output file

No output file; the tagged text (i.e., tokens followed by their part-of-speech tag, one sentence per line) are printed to stdout, from where standard Unix shell operation can write the results to a file.

3.3 Approximation of the time necessary to process the test input file.

Not applicable.

4. ADMINISTRATIVE INFORMATION

4.1 Contact person

Name: Tiago Tresoldi
Address: Universidade Federal do Rio Grande - FURG
Programa de Pós-Graduação em Letras
Av. Itália, Km 08
96203-900 - Rio Grande - RS
Affiliation: Universidade Federal do Rio Grande (FURG)
Position: Ph.D. student in History of Literature
Telephone: +55.53.3233.6614 (Secretaria do Programa de Pós-Graduação em Letras da FURG)
Fax:
E-mail: tresoldi@gmail.com

5. LICENSE

6. RELEVANT REFERENCES AND OTHER INFORMATION

Adwait Ratnaparkhi (1998). *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Eric Brill (1993). “Automatic grammar induction and parsing free text: A transformation-based approach”. In *Proceedings of the 31st Annual Meeting of the ACL*.

Ingo Schröder (2002). “A Case Study in Part-of-Speech tagging Using the ICOPOST Toolkit”. Technical report FBI-HH-M-314/02. Department of Computer Science, University of Hamburg.

Lawrence R. Rabiner (1990). “A tutorial on hidden markov models and selected applications in speech recognition”. In Alex Waibel & Kai-Fu Lee (ed.), *Readings in Speech Recognition*. Morgan Kaufmann, San Mateo, CA, USA, pp. 267-290.

Thosrten Brants (2000). “TnT - as statistical part-of-speech tagger”. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, WA, USA.

Walter Daelemans, Jakub Zavrel, Peter Berck & Steven Gillis (1996). “MBT: A memory-based part of

speech tagger-generator". In Eva Ejerhed & Ido Dagan (ed.), *Proceedings of the Fourth Workshop on Very Large Corpora*, pp. 14-27.