Running Grail Light

Richard Moot

15 January 2014

1 License

2 Acknowledgments

The core engine of the chart parser is a straightforward adaptation of the deductive parser of (Shieber, Schabes & Pereira 1995).

The file `annodis.pl` has been semi-automatically extracted from the Sequoia Treebank (Candito & Seddah 2012) (the formulas have been hand-corrected by me). The Sequoia corpus is released under the GNU Lesser General Public License for Linguistics Resources (LGPL-LR).

Grail Light contains several key files of the Grail parser as well (Moot 1998, Moot 2010).

All files have been designed to run with SWI Prolog, though porting it to other Prologs shouldn't be extremely hard.

3 Installation

Enter the Grail Light directory and type.

```
./configure
```

followed by

```
make all
```

This creates the executable files.

4 Getting started

Grail Light is a chart parser for multimodal categorial grammars designed to work with the Type-logical Treebank, but it can be adapted to work with other corpora as well.

There are two main executables: `grail_light.pl` and `grail_light_cr.pl`, both in the main directory; the only difference between the two files is that the first expects probabilities as input (of which it compute the maximum) whereas the second computes the parse which has the minimum number of crossing with respect to a given set of spans (in the file `annodis.pl`, these spans have been precomputed).

Both files are Prolog scripts, this means they can be run interactively from Prolog or run directly from the command line (with an optional grammar file argument).

For example, if the archive has been unpacked in the directory `my/path`, the following command will run Grail Light on all Annodis sentences from the Sequoia treebank (depending on your computer, this may take some time!).

```
/my/path/grail_light_cr.pl /my/path/annodis.pl
```

If you prefer to take things one sentence at a time, enter the Grail Light directory, start SWI Prolog and load Grail directly as follows (the lines starting with "`%`" are comments and "`?-`" is the Prolog prompt).

```
% Load Grail Light
?- [grail_light_cr].

% Load Annodis sentences
?- [annodis].

% Parse sentence 2
?- sentence(2, Semantics).
```

We first load Grail Light, then the sentences from Sequoia/Annodis and finally, we parse the sentence using the predicate `sentence` which takes a sentence number from the corpus as argument and returns it semantics as produced by the chart parser.

5 LaTeX output

Both the proof produced by the chart parser and its semantics are output as LaTeX files. The file `latex_proofs.tex` contains the chart proofs and the file `semantics.tex` the semantics.

The file `latex_proofs.tex` is just a header file, if you want to copy and paste proofs for use in your own documents, you will find them in the file `proof.tex`.

Since the proofs for even medium-length sentences tend to occupy a lot of horizontal space, `latex_proofs.tex` sets up a very large page. This means you

will probably have to zoom in quite a bit before you have anything readable on your screen!

6 Going further

Parsing you own sentences takes a bit of effort. The predicate `prob_parse` takes as its argument a list of tuples of the form `si(Word, POStag, Lemma, List)`, where `Word` is the word as it occurs in the sentence, `POStag` is its Part-of-Speech tag, `Lemma` is the lexical lemma — eg. "être" (*to be*) for "est" (*is*) — where and `List` is a list of `Formula-Weight` pairs.

The file `annodis.pl` contains a large number of examples.

The file `chart_parser.pdf` (in this directory) gives a more detailed description of the underlying workings of the chart parser.

References

Candito, M. & Seddah, D. (2012), Le corpus Sequoia : Annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical, *in* 'Proceedings of Traitement Automatique des Langues Naturelles (TALN)', Grenoble.

Moot, R. (1998), Grail: an automated proof assistant for categorial grammar logics, *in* R. Backhouse, ed., 'Proceedings of Calculemus/User Interfaces for Theorem Provers', pp. 120–129.

Moot, R. (2010), Wide-coverage French syntax and semantics using Grail, *in* 'Proceedings of Traitement Automatique des Langues Naturelles (TALN)', Montreal.

Shieber, S., Schabes, Y. & Pereira, F. (1995), 'Principles and implementation of deductive parsing', *Journal of Logic Programming* **24**(1–2), 3–36.