# **Assignment #2: Extracting noun groups using manually written rules**

Objectives

The objectives of this assignment are to:

* Write phrase-structure rules to describe partial syntactic structures
* Understand the principles of the DCG parsing mechanisms
* Know what syntactic chunking is
* Optionally, read a short text on an industrial system

Organization and location

The second lab session will take place on

* Group 1: Wednesday, September 16 from 10:15 to 12:00 in the Alpha room
* Group 2: Wednesday, September 16 from 10:15 to 12:00 in the Beta room
* Group 3: Wednesday, September 16 from 13:15 to 15:00 in the Beta room

You can work alone or collaborate with another student.

Each group will have to:

* Write a set of grammar rules in Prolog to detect noun groups.
* Evaluate the results and comment them briefly

Preparation

1. Make sure that you understand what shallow syntax is, and what is the point of it. See page 291-292 in the book, 2nd edition, or take a look at the *Chunk stylebook* from Steven Abney [[pdf](http://www.vinartus.com/spa/96i.pdf)].
2. The lab will use the data from the [CoNLL-2000 shared task](http://www.cnts.ua.ac.be/conll2000/chunking)on chunking. Have a look at the web page and make sure that you understand how the groups are represented in the tabular format (see**Software and Data**).

Programming

Choosing a training and a test sets

1. As annotated data and annotation scheme, you will use the data available from [CONLL 2000](http://www.cnts.ua.ac.be/conll2000/chunking/).
2. Download both the [training](http://www.cnts.ua.ac.be/conll2000/chunking/train.txt.gz)and [test](http://www.cnts.ua.ac.be/conll2000/chunking/test.txt.gz) texts. Decompress them using gunzip.   
   (Local copies available here: [[train.txt](http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/conll2000/train.txt)] [[test.txt](http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/conll2000/test.txt)].)
3. Be sure that you have a running Prolog system. In the computer room, you will use [SWI Prolog](http://www.swi-prolog.org/).

Annotating the groups

1. The two main syntactic groups are the noun groups and the verb groups, also called noun chunks and verb chunks. In the program you will write, you will first consider the noun groups.
2. Find a couple of sentences in English, from a newspaper for instance, and tag manually some noun groups using the tags<ng> and </ng> to delimit the groups.
3. Read the annotation scheme used by CoNLL and transform your tags using the IOB2 annotation scheme. (Notice the difference between the IOB and IOB2 tagging schemes.)
4. Read the convert\_brackets\_to\_iob2/1 predicate in the brackets\_to\_iob2.pl file and experiment it with thetest\_conversion\_iob2/0 predicate. [[Program folder](https://github.com/pnugues/ilppp/tree/master/programs/labs/chunking)].   
   You will start the Prolog interpreter with the command:swipl. Prolog prints a prompt: ?- and reads your commands. You will type:

?- [brackets\_to\_iob2].

to load the file and

?- convert\_brackets\_to\_iob2.

to run the predicate.

Writing rules to detect the noun groups

1. Read the rules describing noun groups from the ng.pl program demonstrated in the lecture on partial parsing and try to understand them. [[Program folder](https://github.com/pnugues/ilppp/tree/master/programs/labs/chunking)]
2. Run the tag\_complete\_file(InputFile, OutputFile) predicate from the group\_detector.pl file. InputFile is either'train.txt' or 'test.txt' and OutputFile is the result from you tagging program.   
   tag\_complete\_file calls the group\_detector(TokenizedList, Result) predicate seen during the lecture on partial parsing. You will also need input/output predicates that you will find in the conll\_io.pl file. You will need to have all these files in your working directory. [[Program folder](https://github.com/pnugues/ilppp/tree/master/programs/labs/chunking)]
3. Measure the performance of the system. Use the [eval.sh](http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/conll2000/eval.sh)script and the [conlleval.txt](http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt)evaluation program used by the CONLL shared task. Run the command:   
   sh eval.sh reference\_file your\_file
4. Examine the rules to detect noun groups and propose improvements. We will clarify this during the laboratory session. You may need to consult a[description of the part-of-speech tags](http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html).
5. Write more rules, and improve the existing rules, iteratively. Evaluate the results on the **training set** to measure your progress. Improve the rules so that you obtain a harmonic mean (FB1) greater than 75 for the noun groups. The applications tkdiff or kdiff3 may be useful when you are inspecting the differences between your output and the correct file.
6. Finally, evaluate the result on the test set.

Writing rules to detect the verb groups

You will now write a program to identify verb groups. There is no accuracy to reach in this laboratory; however you can easily obtain scores better than 50.

1. Use the tags <vg> and </vg> to delimit the groups on some sentences you will collect from newspapers.
2. Use your program and write rules detecting the verb groups.
3. Do you think a perfect detection is possible? Propose improvements.

Complement

Read the text *How Fastus Works* about the [FASTUS](http://www.ai.sri.com/natural-language/projects/fastus.html) system [[pdf](http://www.ai.sri.com/~israel/fastus-schabes-Discern.pdf)]. You can additionally read the [tutorial](http://www.ai.sri.com/~appelt/ie-tutorial/) on information extraction by Douglas Appelt or some of Steven Abney's[publications](http://www.vinartus.com/spa/publications.html).