Aidan Finn

8th July 2004

ELIE is a tool for adaptive information extraction. It also provides a number of other text processing tools e.g. POS tagging, chunking, gazeteer, stemming. It is written in Python.

1 Installation

Requirements:

- Python 2.1 or higher
- Java 2 or higher
- Weka (included in distribution)
- Brilltag (if you intend to use datasets other than those provided)

Unzip the Elie archive. Edit the *basedir, BRILLTAGPATH* and *java* in the file *config.py* to describe your own system. Add *$ELIEHOME/lib/weka.jar* to your java classpath.

2 Usage

Elie contains the following executable files:

- evaluation.py The main way to run ELIE
- scorer.py Calculate performance measures from ELIE logs
- extractor.py Performs basic learning and extraction
- preprocessCorpus.py preprocesses a corpus of text files
- tagging.py does POS, chunking etc on a text file

Execute these files without any arguments to get usage information.

2.1 Input format

Documents should be stored in text files with one document per text-file. Fields should be marked using the syntax <*field*> ... </*field*>.

2.2 Preprocessing

This stage adds tokenisation, orthographic, POS, chunking and gazetteer information to the input files and stores it using an ELIE internal format. This stage only needs to be done once for each document collection! Running ’*preprocessCorpus.py datasetDirectory*’ will create a new directory called *datasetDirectory.preprocessed* which contains all the files in ELIEs internal format.

Note the input files shouldn’t contain any unusual control characters and for every <field> there must be a corresponding </field>.

2.3 Running

The recommended way to run ELIE is using the file *evaluation.py*. It takes the following parameters.

```
-f field
-t trainCorpusDirectory
-D dataDirectory
[-T testCorpusDirectory]
[-s splitfilebase]
[-mpnvh]
If -t and -T are are set, then we train on trainCorpusDirectory and test on
Options:
-m use cached models (NotYetImplemented)
-p set train proportion default=0.5
-n number of trials default=10
-v version info
-h help
```

The corpora directories should contain preprocessed files only i.e. those created by preprocessCorpus.py. The dataDirectory is where ELIE will store all its intermediate and output files. The splitfilebase argument can used be for predefined splits.

3 Output

The detail of ELIEs printed output is controled using the parameter *config.verbosity*.

ELIE produces several logfiles that can be used by the bwi-scorer or ELIEs own scorer (scorer.py). These are located in the specified dataDirectory.

e.g. scorer.py elie.speaker.*.elie.L1.log