Natural Language Processing (NLP)

The Basics

Greta Franzini
Università Cattolica del Sacro Cuore, Milan, Italy
EnExDi Winter School, Poitiers, 9-11 January 2019
Course objectives

By the end of the day, you will:

1. Be able to make use of the command-line
2. Understand and perform basic text analysis tasks
Course format

**Morning** (3:00 hours)
Alternation between theory and practice on a provided French text.

**Afternoon** (2:00 hours + 1 hour OPTIONAL)
Practice on your own data/text/corpus:
- Prepare the texts;
- Identify tasks that can be done here and those that you can do at home.
Command-line
Command-line

Definition
The command-line is a console or user interface to issue commands to a computer’s operating system. The command processor or language of the command-line is Bash.

Open your command-line now:
- **Windows**: Start > Program files > Accessories > Command Prompt
- **Mac OSX**: Applications > Terminal
- **Linux**: Applications > Terminal

N.B. Windows and Mac OSX/Linux use different commands... 😞
Command-line

Why learn the command-line?

- Some **text analysis tools rely on it** to work (e.g., TreeTagger, LEMLAT, TRACER, etc.);
- **software-free data analysis** and **preparation** (e.g., cleaning, removing XML tags, etc.);
- **monitor** running **processes** on a machine;
- **server-side** tasks (e.g., copying a file from your computer to a server);

and much more!
Command-line - PRACTICE

#1: navigate the file system.

In your command-line, type:

- `pwd` Present Working Directory
- `cd PATH/TO/FILE` Change Directory
- `cd ..` Parent directory
- `ls -l` List items in directory
- `mkdir FOLDERNAME` Create (make) a directory
- `mv oldname.txt newname.txt` Rename a file or folder (move)
- `rm file.txt rmdir foldername` Delete (remove) file and delete folder
- `cp filename.txt foldername` Copy a file to another location
- `clear` Clear the screen

[See windows-vs-mac-command.pdf for the Windows vs. Mac/Linux command mapping].
Text formats

Most interoperable file formats for text processing:

- **TXT**: unstructured raw text file.
- **CSV (comma separated values)**: tabular format, i.e., database table or spreadsheet data.
- **TSV (tab separated values)**: tabular format, i.e., database table or spreadsheet data.

Other formats also possible (e.g., XML), but more expensive to (computationally) process (specific parsers) and less interoperable.
Command-line - PRACTICE

#2: transform XML to raw text (TXT) using regular expressions (regex)

In your command-line:

- Navigate to the folder where you saved MOLIERE_MISANTHROPE.xml: `cd PATH/TO/FOLDER`
- Open MOLIERE_MISANTHROPE.xml: `cat MOLIERE_MISANTHROPE.xml`
- Remove all text enclosed in angle brackets: `cat MOLIERE_MISANTHROPE.xml | sed \'s/\<[^<>]*\>/\//g\'`
  - You must use the pipe | to concatenate tasks.
  - sed: Stream EDitor; powerful command typically used for text replacement.
  - \'s///g\' s = substitute; / = delimiter; g = global.
- Save the XML-free text as TXT: `cat MOLIERE_MISANTHROPE.xml | sed \'s/\<[^<>]*\>/\//g\' > MOLIERE_MISANTHROPE.txt`
Other useful commands.

- \texttt{grep 'word' filename} to extract all instances of a word in a file.
- \texttt{egrep 'hello there' filename} to search sentences containing ‘hello there’.
- \texttt{tail filename} to return the last 10 lines of the file.
- \texttt{tail -20 filename} to return the last 20 lines of the file.
- \texttt{head filename} to return the first 10 lines of the file.
- \texttt{head -20 filename} to return the first 20 lines of the file.
- \texttt{top} to view all running processes on a machine.
- \texttt{df "Disk Free"}, to check storage space in the directory.
Text Analysis
Data pre-processing

“It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data (Dasu and Johnson 2003). Data preparation is not just a first step, but must be repeated many times over the course of analysis as new problems come to light or new data is collected.” (Hadley Wickham, 2014)

http://vita.had.co.nz/papers/tidy-data.html
Levels of text analysis

1. **Tokenisation** (segmentation)
2. **Grammatical analysis** (*Part-of-Speech tagging*)
3. Lemmatisation
4. Morphological analysis
5. **Syntactic analysis** (*parsing*)
1. Tokenisation (segmentation)

Definition
Act of breaking a string or sequence of strings into tokens, typically words but also numbers, punctuation, symbols, acronyms, etc. Essential pre-processing task for any lexical analysis.

- The cat is under the table

Token vs. Type
- **Token** = occurrence of a word
- **Type** = unique form of a word

- The cat is under the table

6 tokens and 5 types
1. Tokenisation (segmentation)

Problems of tokenisation

Open https://text-processing.com/demo/tokenize/ and type “Bienvenue à l'école d'hiver”. Observations?

● Spaces and punctuation
  ○ Alors, / (et / ou)
  ○ Character sequences corresponding to multiple tokens without white-space
  ○ L'homme / Milan-Rome

● Acronyms, dates, abbreviations, multi-word expressions (MWE)
  ○ U.S.A. / 05.02.2019 / Mr. / New York / ad hoc

How to tokenise?

● Command-line;
● scripts (Python, Java, etc.);
● tokensisers.
Tokenisation (segmentation) – PRACTICE

#1: calculate the Type-Token Ratio (TTR) or *lexical variance*.

In your command line:
- Open `MOLIERE_MISANTHROPE.txt`
- Transform all upper case characters to lowercase: `cat MOLIERE_MISANTHROPE.txt | tr '[:upper:]' '[:lower:]'`
- Transform all punctuation into new lines: `cat MOLIERE_MISANTHROPE.txt | tr '[:upper:]' '[:lower:]' | tr '[:punct:]' '
'`
- Transform all spaces into new lines: `cat MOLIERE_MISANTHROPE.txt | tr '[:upper:]' '[:lower:]' | tr '[:punct:]' '
' | tr '[:space:]' '
'`
- Remove all blank lines: `cat MOLIERE_MISANTHROPE.txt | tr '[:upper:]' '[:lower:]' | tr '[:punct:]' '
' | tr '[:space:]' '
' | sed '/^\s*$/d'`
- Save results as a new file entitled `MOLIERE_MISANTHROPE.txt.tokens`
- Count the number of lines in `MOLIERE_MISANTHROPE.txt.tokens`
Tokenisation (segmentation) – PRACTICE

#1: calculate the Type-Token Ratio (TTR) or *lexical variance*.

- Open `MOLIERE_MISANTHROPE.txt.tokens`
- Sort the tokens alphabetically: `cat MOLIERE_MISANTHROPE.txt.tokens | sort`
- Remove duplicates: `cat MOLIERE_MISANTHROPE.txt.tokens | sort | uniq -c`
- Sort again by frequency (first column): `cat MOLIERE_MISANTHROPE.txt.tokens | sort | uniq -c | sort -k1nr`
- Save results as a new file entitled `MOLIERE_MISANTHROPE.txt.types`
- Count the number of lines in `MOLIERE_MISANTHROPE.txt.types`

**Type-Token Ratio: (Types /Tokens) * 100 = N% = Lexical variance/richness**

The more types there are in comparison to the number of tokens, then the more varied is the vocabulary (the higher the %, the higher the lexical variance)
2. Grammatical analysis: PoS-tagging

- **Parts of Speech**: noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction and interjection (English).
- **A word can have more than one PoS**, e.g., homographs: *close* (verb or adverb), *bear* (verb or noun), *part* (verb or noun), etc.
- **PoS-tagging** (PoS-disambiguation) is the practice of assigning the correct PoS to words.
- There are many PoS-taggers, each using a different set of tags (Penn tagset, Universal Dependencies PoS tags, etc.). A tag-set can have up to 200 tags! Problem of interoperability between tag-sets.
2. Grammatical analysis: PoS-tagging

Methods:

- **Rule-based** (intuition-based; supervised): predetermined, arbitrary rules that the machine has to follow.
  - Language dependent
  - In heavy use until early 90s

- **Data-driven** (empirical, statistical; unsupervised): the machine learns the rules from empirical evidence.
  - Language independent
  - In use since the second half of the 90s
  - Relies on linguistic resources and annotated data

- **Mixed approach**
  - TreeTagger
3. Lemmatisation

- Reduces a *word form* to its *lemma* (dictionary entry)
  - *wanted, wants* $\rightarrow$ *want (V)*

- Morphological ambiguity (PoS-tagging)
  - *Close* $\rightarrow$ *close (V) | Close $\rightarrow$ *close (ADV) | Close $\rightarrow$ *(Glenn) Close (N)*

- Many lemmatisers, different accuracy
  - Go to *LemmaGen* at [http://lemmatise.ijs.si](http://lemmatise.ijs.si), select ‘French’, and try lemmatising different French sentences. Observations?
Download **TreeTagger** and the French parameter file
- URL: [http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/](http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/)
- Place all of the downloaded files into a folder on your Desktop entitled `treetagger`

Place a copy of `MOLIERE_MISANTHROPE.txt` in the `treetagger` folder

To run **TreeTagger** on `MOLIERE_MISANTHROPE.txt` (TreeTagger tokenises for you!):
- Open your command-line
- Using `cd`, navigate to the `treetagger` folder on your Desktop
- Once you’re in the `treetagger` folder, type:

  ```
  cat MOLIERE_MISANTHROPE.txt | cmd/tree-tagger-french > MOLIERE_MISANTHROPE.txt.tagged
  ```

Open `MOLIERE_MISANTHROPE.txt.tagged` in Sublime Text Editor. Any unknown words? Any errors?

Using the command-line, extract all unknown words from `MOLIERE_MISANTHROPE.txt.tagged` and save them in a file called `MOLIERE_MISANTHROPE.txt.unknown`
Using the command line, open the **MOLIERE_MISANTHROPE.txt.tagged** and:

- count the number of lines;
- sort the list alphabetically;
- put everything in lower case;
- delete duplicates and count the number of lines;
- replace punctuation with new lines;
- sort by frequency (first column).

What are the most frequent words in the text?
4. Morphological analysis

Assigns **morphological information** to word forms:
- PoS tags
- tense, voice, mood, number, gender, person, case, etc.

Word Formation Latin

LEMLAT 3
5. Syntactic analysis/parsing

- *To parse* = “to divide (a sentence) into grammatical parts and identify the parts and their relations to each other. (Merriam-Webster)”.
- Parsers rely on (manually) annotated data, often treebanks.
- **Treebank** = syntactically-annotated corpus:
  - Lemmatisation (disambiguated)
  - Morphological features (disambiguated)
  - Syntax
- Two types of treebank:
  - **Constituent**: phrase structure
  - **Dependency**: dependency structure
5. Syntactic analysis/parsing

https://en.wikipedia.org/wiki/Treebank
5. Syntactic analysis/parsing

French corpus of 10M words and treebank freely available at:
https://www.ortolang.fr/market/corpora/cefc-orfeo
Voilà

Greta Franzini
Università Cattolica del Sacro Cuore, Milan, Italy
EnExDi Winter School, Poitiers, 9-11 January 2019
Tools

- TXM: http://textometrie.ens-lyon.fr/?lang=en
- Voyant Tools: https://voyant-tools.org/?lang=fr
- Stanford Core NLP: https://stanfordnlp.github.io/CoreNLP/
- spaCy: https://spacy.io/
- French Wordnet: https://wonef.fr/try/
- OpenNLP: https://opennlp.apache.org
- Corpus-tools.org: http://corpus-tools.org/home/
- TextAnalysisOnline: http://textanalysisonline.com/
- LemmaGen: http://lemmatise.ijs.si/Services
- CATMA: http://catma.de/
- Orange Text Mining: https://orange.biolab.si/
- Open Parallel Corpus: http://opus.nlpl.eu/
Tutorials

- Basic Linux commands: http://www.hongkiat.com/blog/basic-linux-commands/
- Bash tutorial: http://guide.bash.academy/
- RegexR: tool to learn and build regular expressions: http://regexr.com/
- Stack Overflow forum: https://stackoverflow.com/

Mailing list

- Corpora: http://clu.uni.no/icame/corpora/