

CLAM

Bringing your NLP command-line tools to the web!



LANGUAGE MACHINES

**CLST | Centre for Language and
Speech Technology**
Radboud University



Introduction

Observation: NLP tools are often command-line programs ... for good reason.



Command line tools: pros

Command-line tools are a good thing!

"This is the Unix philosophy: Write programs that do one thing and do it well. Write programs to work together." (Doug McIlroy)

- **Flexibility & Extensibility:** Integrate tools into pipelines, the output of one tool is the input to another
- **Performance:** Little overhead
- **Modularity:** Separate the interface from the actual program

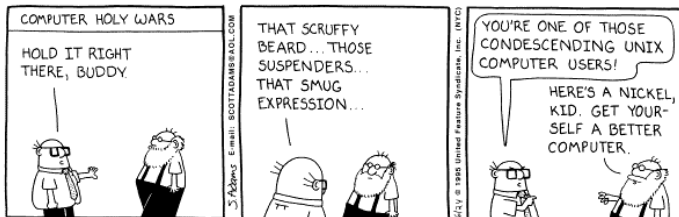


Command line tools: pros

Command-line tools are a good thing!

"This is the Unix philosophy: Write programs that do one thing and do it well. Write programs to work together." (Doug McIlroy)

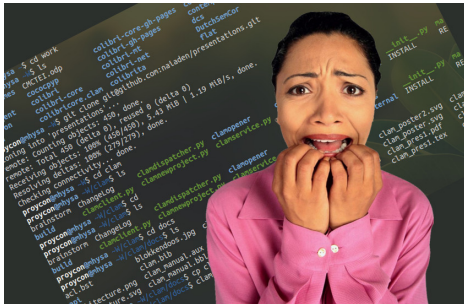
- **Flexibility & Extensibility:** Integrate tools into pipelines, the output of one tool is the input to another
- **Performance:** Little overhead
- **Modularity:** Separate the interface from the actual program



Command line tools: cons

But..

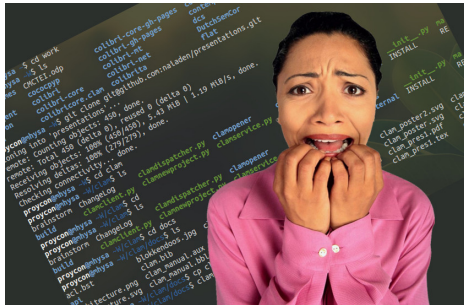
- The command-line is challenging and intimidating for non-technical end-users



Command line tools: cons

But..

- The command-line is challenging and intimidating for non-technical end-users



- Installation may be complex and depend on other software
- Web-connectivity has to be explicitly built-in in your program (not trivial)

CLAM as a solution

What is CLAM?

CLAM is software that wraps itself around your command-line NLP-tool and:

- Offers an automatically generated **web-based user-interface** for human end-users to interact with your tool
- Offers an automatically generated **RESTful webservice** interface for automated clients to interact with your tool

How to use CLAM?

You can wrap your application with minimal effort:

1. .. write a **service configuration** specifying what kind of input your program expects and what output it produces. The interfaces can be generated on the basis of this.
2. .. write a **wrapper script** that acts as the glue between CLAM and your tool

Text Statistics (CLAM Demo)

test

Status

Accepting new input files and selection of parameters

Abort and delete project

Input

Input files

Show 10 entries

Search:

Input File	Template	Format	Actions
test.txt	Input text document		

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Upload a file from disk

Use this to upload files from your computer to the system.

Step 1 First select what type of file you want to add:

Step 2 Set the parameters for this type of file:
Select a type first

Step 3

Grab a file from the web

Retrieves an input file from another location on the web.

Step 1 First select the desired input type:

Step 2 Set the parameters for this type of file:
Select a type first

Step 3 Enter the URL where to retrieve the file

Step 4

Add input from browser

You can create and add new files from within your browser:

Parameter Selection

Main

Create Lexicon

Generate a separate overall lexicon?

☐

Case Sensitivity

Enable case sensitive behaviour?

Limit frequencylist

Limit entries in frequencylist to the top scoring ones. Value of zero (no limit) or higher

Author

Sign output metadata with the specified author name

Powered by CLAM - Computational Linguistics Application Mediator
by Maarten van Gorpel
Induction of Linguistic Knowledge Research Group, Tilburg University



Typical workflow

1. User (or automated client) creates a project
2. User uploads input files
3. User sets parameters for the run
4. User presses the “START” button
5. The tool runs for a certain time (may be long), progress status is reported back to the user
6. When done, the output files are presented
7. User may select output files for viewing or download



Notable Features

- Optimised for **batch processing** and dealing with **large files**, your tool may run for hours or days if necessary
- **Storage model**: files are uploaded and downloaded, they stay on server in “projects” until explicitly removed.
- Extensive **user-authentication** support (including OAuth2).
- Extensive support for **metadata** and **provenance data**
- Suitable for use in external workflow management systems.
- Support for quick real-time “actions”; tie scripts to URLs.
- Support for viewers and convertors
- **Python API** for Python users (clients & service providers)
- Used by various projects in **CLARIN-NL** and others (CLAM is funded through CLARIN-NL)



Demo

- CLAM website: <http://proycon.github.io/clam>
- Numerous webservices from our department are hosted here:
<http://webservices-lst.science.ru.nl>
- (register for a free account if you have none yet)

