TICCLops = TICCL + CLAM

Martin REYNAERT and Maarten VAN GOMPEL

Tilburg centre for Cognition and Communication - ILK

Tweede CLARIN-NL Bijeenkomst, Utrecht. 28 October, 2010
TICCL for TYPOS and OCR-errors

- Tool to perform large scale, unsupervised spelling correction of corpora
- Based on indexing with Anagram Hashing
- Spelling correction = reduction of lexical variation caused by typos, OCR-errors, historical orthographical changes...

- Lexical variants are linked to their most likely canonical form
- Output: lists and ’enriched’ OCR-ed texts
- Production version developed according to KB specifications, second half 2008

Title: Character confusion versus focus word-based correction of spelling and OCR variants in corpora

Author: Martin Reynaert

CLARIN-NL acknowledgement

Online publication: next week, in Open Access
TICCL online processing system

- A demonstration project which will allow CLARIN users to submit their corpora for fully automatic spelling correction and normalization by TICCLops, the online processing version of our core component TICCL. This system should be widely applicable in all manner of curation projects and lexicographical work.
- Start Date: 1 February 2010
- End Date: 31 July 2010
TICCLops: PARTNERS

- Coordination & Technology Provider:  
  Tilburg centre for Creative Computing (TiCC) - Tilburg  
  - Martin Reynaert: Researcher  
  - Maarten van Gompel - Scientific Programmer

- User & Data Provider:  
  National Library (Koninklijke Bibliotheek - KB) - The Hague  
  - Astrid Verheusen - Head Digitisation Department

- CLARIN Center & Data Provider:  
  Institute for Dutch Lexicology (INL) - Leiden  
  - Remco van Veenendaal - Head TST-Centrale
Research Data

- Staten-Generaal Digitaal: 180 years of OCR-ed Acts of Parliament and related documents, i.e. in historical and contemporary spelling. (http://www.statengeneraaldigitaal.nl)
- Database of Digitized Daily Newspapers: 8 million pages, goes back to 1618. (http://kranten.kb.nl).
- Collections of recently digitized copyright-free books and magazines (forthcoming: Google Books: 160,000 books)
TICCLops: TICCL online processing system

- In the course of the project we developed a more generic solution than initially proposed:
  Computational Linguistics Application Mediator: CLAM

- TICCLops and CLAM form the basis for several work packages in the Dutch-Flemish CLARIN project TTNWW (tokenization, text conversion, POS-tagging, lemmatization, NER, shallow parsing)

- CLAM was made available to other projects and was adopted by Call 1 project ADELHEID
Observation

There are a lot of specialised command-line NLP tools available.

Problems

1. Tools often available only locally, installation and configuration can be tough
2. Not very user-friendly for the untrained general public or technically-challenged researchers
3. How to connect one tool to another?
Solution
Making NLP tools available as full-fledged webservices.

Advantages

1. Services available over the web.
2. User-friendly interface built-in in the webservice
3. Great for demo purposes
4. Greater for only providing access to what is fit for general use
5. Multiple webservices can be chained in a workflow
Our Focus

1. A *universal* approach: *wrapping*
   - Turn almost *any* NLP tool into a webservice with *minimal effort*
   - NLP tool = Given input files and a custom set of parameters, produce output files
   - No need to alter the tool itself

Wrapping Approach

1. NLP application: blackbox
2. Wrapper script
3. CLAM Webservice
Technical Details

RESTful Webservice

RESTful Webservice (as opposed to SOAP, XML-RPC)

1. Resource-oriented: "Representations" of "resources" (projects)
2. Using HTTP verbs
3. Lightweight
4. Returns human-readable, machine-parseable XML adhering to a CLAM XML Scheme Definition
5. User authentication in the form of HTTP Digest Authentication
Python

Written entirely in Python 2.5

1. NLP tools, wrapper scripts, and clients may be in any language
2. But: Readily available API when writing wrapper scripts and clients in Python.
3. Integrated into Apache for production work, lightweight solutions also available
Built-in User Interface

User interface automatically generated from XML using XSLT (in browser)

1. Webservice *directly* accessible from webserver
2. Web 2.0 interface: xHTML Strict, jquery (javascript), AJAX, CSS
CLAM Setup

Projects are the main resources, users start a new project for each experiment/batch.

**Three states:**

- **Status 0)** Parameter selection and file upload
- **Status 1)** System in progress
  - Actual NLP tool runs at this stage only
  - Users may safely close browser, shut down computer, and come back later in this stage
- **Status 2)** System done, view/download output files
Providing a Service

In order to make a webservice:

1. Write a service configuration file (in Python, but no Python experience required).
   - General meta information about your system (name, description, etc..)
   - Definition of parameters accepted by your system/wrapper script
   - Definition of input formats and output formats
   - Definition of users and authentication method

2. Write a wrapper script for your system
   - Wrapper script is invoked by CLAM, and should in turn invoke the actual system
   - Acts as glue between CLAM and your NLP Application.
   - Can be written in any language (python users may benefit from the CLAM API)
   - Not always necessary, NLP applications can be invoked directly by CLAM as well.
TICCLops: the user experience

TICCLops: Start-up screen
TICCLops: the user experience

TICCLops: Corpus Upload

Status
Ready to start

Input
Input source: Use uploaded files ▼
Input files

Show 10 ▼ entries

<table>
<thead>
<tr>
<th>Input File</th>
<th>Format</th>
<th>Encoding</th>
<th>Actions</th>
</tr>
</thead>
</table>

No matching records found

Showing 0 to 0 of 0 entries

Upload a file from disk

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

Step 1) First select the desired input format for this upload: Koninklijke Bibliotheek XML-formaat [UTF-8] ▼
Step 2) Select and upload a file

Grab a file from the web

Retrieves an input file from another location on the web.

Step 1) First select the desired input format: Koninklijke Bibliotheek XML-formaat [UTF-8] ▼
Step 2) Enter the URL where to retrieve the file ▼
Step 3) Retrieve and add file

Add input from browser

You can create new files right from your browser: Open Live Editor

Parameter Selection

Lexicon
Change a Quick Lexicon

Done
TICCLops: Parameter Selection

Parameter Selection

Lexicon

Choose a Dutch Lexicon
Dutch Lexicon

Focus Word Selection

Minimum Word Frequency
Integer between zero and ten million

Maximum Word Frequency
Integer between zero and ten million

Minimum Word Length
Integer between zero and one hundred

Maximum Word Length
Integer between zero and one hundred

Edit/Levenshtein Distance

How many edits?
Search N characters for variants
Only 1 edit

Basic name for Output Files

File name
Enter a basic file name

N-best Ranking

How many ranked variants?
Return N best-first ranked variants
First-best Only

Options

Evaluation
Conversion
Debugging

Start
TICCLops: Progress Monitoring

Status
Primary Loop: FOCUS WORD PROGRESS: at word number 292 of 936

31%

Abort and delete project

You may safely close your browser or shut down your computer during this process, the system will keep running and be available when you return another time.

Primary Loop: FOCUS WORD PROGRESS: at word number 292 of 936
Primary Loop: FOCUS WORD PROGRESS: at word number 282 of 936
Primary Loop: FOCUS WORD PROGRESS: at word number 273 of 936
Primary Loop: FOCUS WORD PROGRESS: at word number 264 of 936
Primary Loop: FOCUS WORD PROGRESS: at word number 254 of 936

Powered by CLIR - Computational Linguistics Application Mediator
by Marleen van Gompel

Induction of Linguistic Transfer Rules Group, Tilburg University
Thanks!!

Thank you for your attention!

Papers about TICCL are available at:
http://ilk.uvt.nl/

TICCLops = TICCL + CLAM

Martin REYNAERT and Maarten VAN GOMPEL

Tilburg centre for Cognition and Communication - ILK

Tweede CLARIN-NL Bijeenkomst, Utrecht. 28 October, 2010