U-Compare Syntactic Chunking Service

1. BASIC INFORMATION

Service name

U-Compare Syntactic Chunking Service

Overview and purpose of the tool

This is a web service that identifies and categorises syntactic chunks in Galician text. Also identifies sentence and tokens, and assigns parts-of-speech and lemmas to tokens.

A short description of the algorithm

This web service is based on a UIMA-based workflow, created using the U-Compare text mining system[1]. The workflow was exported from U-Compare as a web service using the built in functionality (Kontonastsios et al., In Press). The workflow was created as part of the work to increase the number of interoperable tools operating on different European languages (Ananiadou et al, 2011).

The workflow consists of a single UIMA compliant tool:

1) Freeling[2] parsed web service (performs syntactic chunking)[3], configured to operate on Galician (service provided by the PANACEA project[4])

2. TECHNICAL INFORMATION

Software dependencies and system requirements

This is a web service that can be run from a browser or accessed programmatically. The only basic requirement is an internet connection.

Installation

There is no installation. The web service can be accessed at the following URL:

http://nactem001.mib.man.ac.uk:8080/UCompareWebServices/SyntacticChunking_FreelingParser

---

[1] http://nactem.ac.uk/ucompare/

[2] http://nlp.lsi.upc.edu/freeling/

[3] http://registry.elda.org/services/205

[4] http://www.panacea-lr.eu/

The web form available at this URL is shown in Figure 1, with some Galician text entered.

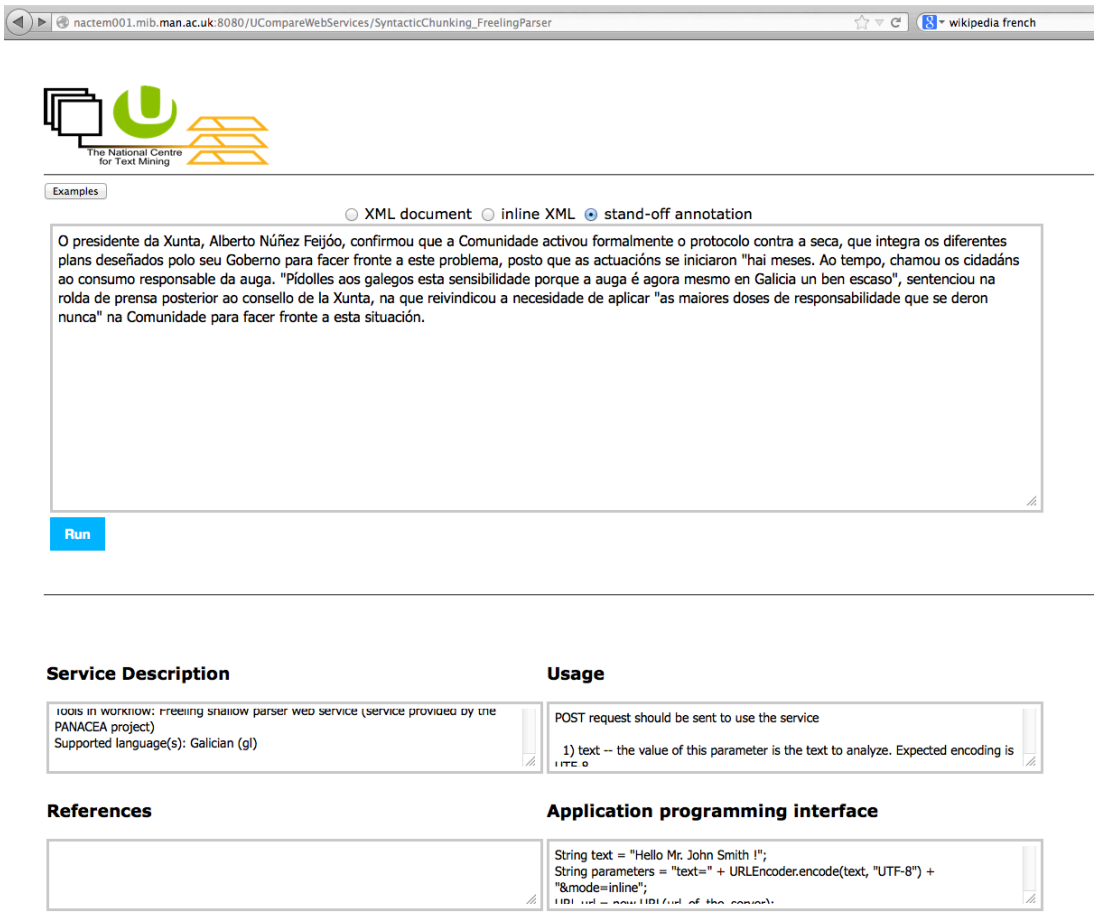

Figure 1: Web form for the service

Execution instructions

The web service can be executed by typing or pasting text into the online form and clicking on the “Run” button.

Alternatively, the web service can executed from within program code, as explained in the “Usage” and “Application programming interface” boxes of the web form.

A POST request should be used to call the service. The following parameters may be used in the request:

- **text** - the value of this parameter is the text to analyze. Expected encoding is UTF-8. This parameter is obligatory.
- **lang** - This parameter sets the language of the text. If this parameter is not provided, then the value"en" will be used

- **mode** - This parameter sets the format of the annotated information returned by the service. If this parameter is not set, XML output will be produced. The two possible types of output are as follows:
    - **inline** – annotations are encoded as inline XML.
    - **xml** – results are output as an XML document containing the annotations added

The following code example shows how the web service can be called from Java code:

```
//Set the input text
String text = " <Text_to_be_analysed>";
//Set the parameter string
String parameters = "text=" + URLEncoder.encode(text,
"UTF-8") + "&mode=inline";
//Create the URL connection
URL url = new
URL("http://nactem001.mib.man.ac.uk:8080/UCompareWebServi
ces/SyntacticChunking_FreelingParser");
URLConnection connection = url.openConnection();
connection.setDoOutput(true);
//Create Output stream
OutputStreamWriter writer = new
OutputStreamWriter(connection.getOutputStream());
//write parameters to output stream
writer.write(parameters);
writer.flush();

//Read the results returned by the service
BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream(), "UTF-8"));
String line;
while ((line = reader.readLine()) != null) {
    System.out.println(line);

}
```

Input/Output data formats

Input data formats

The input is plain text, UTF-encoded.

Output data format

If the service is run from the web interface, then the output is visualized in the interface using colored highlights in the text to show the individual annotations, and one or more tables of information below, each corresponding to a particular type of annotation.

If the service is run programmatically, then the output is provided in XML format. See section 3 for an example.

Integration with external tools

The API allows the functionality of the web service to be embedded in any application.

3. CONTENT INFORMATION

Using the web interface, the output of the service is visualised as shown in Figure 2.

para	SPS00	para
facer	VMN0000	facer
fronte a	CA	fronte_a
esta	DD0FS0	este
situación	NCFS000	situación
.	Fp	.

Chunk	labelString
O presidente da Xunta, Alberto Núñez Feijóo, confirmou que a Comunidade activou formalmente o protocolo contra a seca, que integra os diferentes plans deseñados polo seu Goberno para facer fronte a este problema, posto que as actuacións se iniciaron "hai meses. Ao tempo, chamou os cidadáns ao consumo responsable da auga. "Pídolles aos galegos esta sensibilidade porque a auga é agora mesmo en Galicia un ben escaso", sentenciou na rolda de prensa posterior ao consello de la Xunta, na que reivindicou a necesidade de aplicar "as maiores doses de responsabilidade que se deron nunca" na Comunidade para facer fronte a esta situación.	grup-verb
O presidente da Xunta, Alberto Núñez Feijóo,	sn
O	j-ms
presidente	n-ms
da Xunta	sp-de
a Xunta	sn
a	j-fs
Xunta	w-fs
,	Fc
Alberto Núñez Feijóo	w-ms
,	Fc
confirmou	verb
que	conj-subord
a Comunidade activou formalmente o protocolo contra a seca, que integra os diferentes plans deseñados polo seu Goberno para facer	grup-verb
a Comunidade	sn
a	j-fs
Comunidade	w-fs
activou	verb
formalmente	sadv

Figure 2: Visualisation of web service output

In Figure 2, the different types of annotations are displayed in different tables. At the top of the screen is the bottom of the table containing information about tokens, their parts-of-speech and lemmas. At the bottom of the screen is the “Chunk” table,

showing all the (possibly overlapping) chunks that have been identified by the service. For each chunk, the span of text is displayed, together with the category assigned to the chunk, in the "labelString" column.

An example of the XML output format, which is more suited to programmatic use, is shown in Figure 3. In the XML, different XML element types encode the different types of annotations. In the example shown,, the RichToken elements store information about tokens, while the Chunk annotations store information about syntactic chunks. In each XML element, the start and end offsets of the corresponding annotation in the text are encoded in the "begin" and "end" attributes. Other attributes are possible, depending on the element type. The Chunk type stores the chunk category in the "labelString" attribute, while the RichToken type stores the part-of-speech tag and base form assigned to each token, in the "posString" and "base" attributes, respectively.

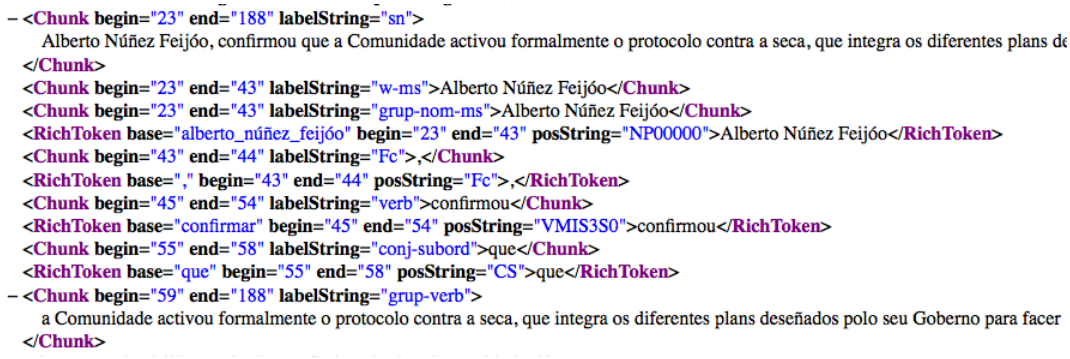

Figure 3: XML output example

3. LICENCE

a) The web service only is licenced NaCTeM Web Service Licence Agreement (standard non-commercial use) – see "U-Compare-Syntactic-Chunking-Service-Licence.pdf" in the "licences" directory. Please contact us using the details below if you require a commercial licence.

b) The tools used in the workflow on which the web service is based may have their own licences. The NaCTeM Web Service Licence Agreement does NOT apply to these tools.

4. ADMINISTRATIVE INFORMATION

Contact

For further information, please contact Sophia Ananiadou:

sophia.ananiadou@manchester.ac.uk

5. REFERENCES

Ananiadou, S., Thompson, P., Kano, Y., McNaught, J., Attwood, T. K., Day, P. J. R., Keane, J., Jackson, D. and Pettifer, S.. (2011). Towards Interoperability of European Language Resources. *Ariadne,* 67.

Kontonatsios, G., Korkontzelos, I., Kolluru, B., Thompson, P. and Ananiadou, S. (In Press). Deploying and Sharing U-Compare Workflows as Web Services. *Journal of Biomedical Semantics.*