

# Text Handling as a Web Service for the IULA Processing Pipeline

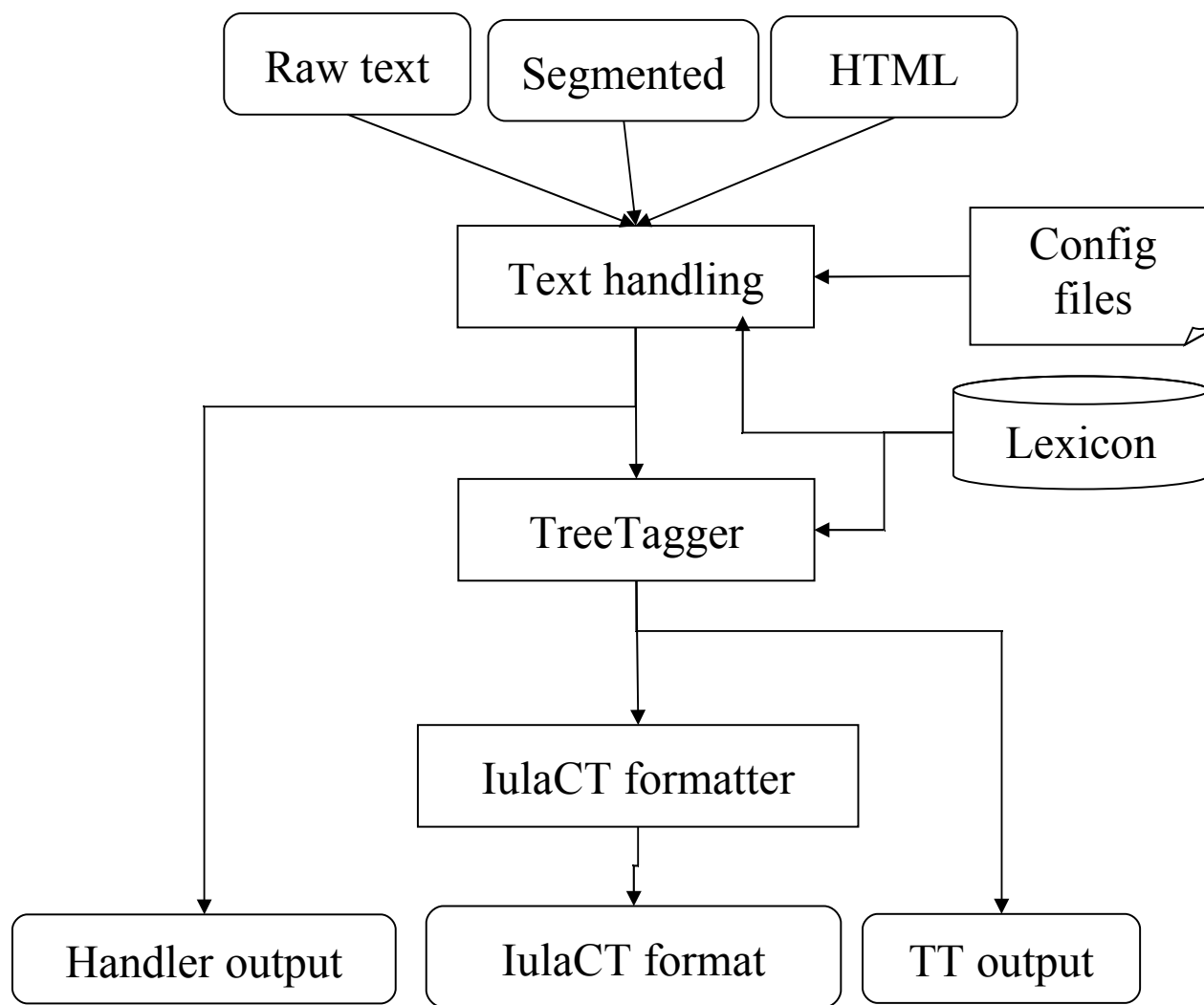
Héctor Martínez, Jorge Vivaldi and Marta Villegas

*Grup de Tecnologies dels Recursos Lingüístics (TRL),  
Institut Universitari de Lingüística Aplicada (IULA)  
Universitat Pompeu Fabra (UPF)*

# Index

- 1 - Text Handling
- 2 - Rationale for a WS
- 3 - Service integration
- 4 - Common Interface
- 5- Conclusions and further work
- 6- Questions

# 1 - Text Handling and Processing Pipeline



# 1- Text Handling features

- Catalan / English / Spanish
- Sentence-boundary detection
- Named entity recognition
- Numerals, Dates, IPs, e-mail addresses...

## 2 – Rationale for a WS

- No need for end-user installation
- Latest version guaranteed
- Access control to internal data (no direct DB access by end user)
- Ease of usage

## 2 - Rationale for a WS - caveats

- Server availability
- Need for connection
- Ease of usage at the expense of throughput
- Security issues - Access control to WS? DoS?

## 3 - Service integration

Deployment of NLP tools as SOAP Web Services

Definition of common interfaces

Definition of shared types

....

## 3 - Service integration

Command line  $\longrightarrow$  WSDL message (SOAP Web Services)

**\$ TagText -text**

**-numlines**  
**-tagonly**  
**-prepronly**  
**-tagblanks**  
**-notagurl**  
**-notagemail**  
**-notagip**  
**-notagdns**  
**-encoding**  
**-errors**

**name="TagText"**

**part name="numlines"**  
**part name="Tagonly"**  
**part name="Prepronly"**  
**part name=" Tagblanks"**  
**part name="notagurl"**  
**part name="Notagemail"**  
**part name="Notagip"**  
**part name="Notagdns"**  
**part name="Encoding"**  
**part name="Errors"**

### 3 - Service integration

```
<wsdl:message name="CommandLineRequest">  
  <wsdl:part name="parameters" element="parameters">  
  </wsdl:part>  
</wsdl:message>
```



#### (Shared) type declaration

```
<wsdl:types>  
  
</wsdl:types>
```

## 4- WS Common Interface

```
<xsd:complexType name="PosTaggerParams">
  <xsd:all><xsd:element name="opt_params" type="xsd:string"/>
  <xsd:element name="main_params" type="typens:MainParams"/>
</xsd:all>
</xsd:complexType>
<xsd:complexType name="MainParams">
  <xsd:all><xsd:element name="text" type="xsd:string"/>
  <xsd:element name="language_code" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
```

## 5 - Conclusions

- Need for common interface
  - Starting with minimal, uncontroversial SW
- Need for stronger typing than “string”

## 6 - Further work for back-end application

- Standoff annotation?
- Expand to French, Italian
- Treebank project → Syntactic parsing as WS

## 6 - Further work - WS

- Text typing not available on WSDL
  - Raw text
  - Tokenized
  - Form + POS + Lemma
  - Other
  - Standoff?

## Accessing the resource

- Web Service
  - <http://kurwenal.upf.edu/WS/hector/service/invoke>
  - <http://kurwenal.upf.edu/WS/hector/service/wsdl>
- Sample demonstration Website
  - <http://melot.upf.edu/cgi-bin/PInCorpus/hectormain.pl>

## Further info and other resources

- CLARIN-Es website
  - <http://clarin-es.iula.upf.edu/>
- IULA-TRL group
  - <http://www.iula.upf.edu/trl/rpreses.htm>

Thank you!