

A ReSTful Web Service for multilingual LRT

Kamel Nebhi

Laboratory of Analysis and Technology of Language
Department of linguistic
University of Geneva

INFuture 2011

Introduction and Motivation

Access to Linguistic Resources and Tools (LRT) is often restricted

- User has to download and install LRT on local machine (updates, new version, patches, ...)
- Developer has to publish and maintain LRT (sustainability, long term support, ...)

How to make available on the Web several tools developed at the LATL of Geneva ?

- Fips, a multilingual parser [3]
- ITS-2, a multilingual translation system [4]
- FipsVox, a french text-to-speech system [1]

Introduction and Motivation

Access to Linguistic Resources and Tools (LRT) is often restricted

- User has to download and install LRT on local machine (updates, new version, patches, ...)
- Developer has to publish and maintain LRT (sustainability, long term support, ...)

How to make available on the Web several tools developed at the LATL of Geneva ?

- Fips, a multilingual parser [3]
- ITS-2, a multilingual translation system [4]
- FipsVox, a french text-to-speech system [1]

Introduction and Motivation

Access to Linguistic Resources and Tools (LRT) is often restricted

- User has to download and install LRT on local machine (updates, new version, patches, ...)
- Developer has to publish and maintain LRT (sustainability, long term support, ...)

How to make available on the Web several tools developed at the LATL of Geneva ?

- Fips, a multilingual parser [3]
- ITS-2, a multilingual translation system [4]
- FipsVox, a french text-to-speech system [1]

Introduction and Motivation

Access to Linguistic Resources and Tools (LRT) is often restricted

- User has to download and install LRT on local machine (updates, new version, patches, ...)
- Developer has to publish and maintain LRT (sustainability, long term support, ...)

How to make available on the Web several tools developed at the LATL of Geneva ?

- Fips, a multilingual parser [3]
- ITS-2, a multilingual translation system [4]
- FipsVox, a french text-to-speech system [1]

Introduction and Motivation

Access to Linguistic Resources and Tools (LRT) is often restricted

- User has to download and install LRT on local machine (updates, new version, patches, ...)
- Developer has to publish and maintain LRT (sustainability, long term support, ...)

How to make available on the Web several tools developed at the LATL of Geneva ?

- Fips, a multilingual parser [3]
- ITS-2, a multilingual translation system [4]
- FipsVox, a french text-to-speech system [1]

Introduction and Motivation

Solution

A **ReSTful Web Service** for accessing all of these technologies simply by using a browser

Web servicization of LRT allow :

- easy access for non-experts through a Web 2.0 interface
- integration of the various different tools and resources in a single environment
- development of Web applications (Learning Management Systems)



Introduction and Motivation

Solution

A **ReSTful Web Service** for accessing all of these technologies simply by using a browser

Web servicization of LRT allow :

- easy access for non-experts through a Web 2.0 interface
- interoperability between different tools and resources [2]
- development of Web applications (eLearning, ...)

Introduction and Motivation

Solution

A **ReSTful Web Service** for accessing all of these technologies simply by using a browser

Web servicization of LRT allow :

- easy access for non-experts through a Web 2.0 interface
- interoperability between different tools and resources [2]
- development of Web applications (eLearning, ...)



Introduction and Motivation

Solution

A **ReSTful Web Service** for accessing all of these technologies simply by using a browser

Web servicization of LRT allow :

- easy access for non-experts through a Web 2.0 interface
- interoperability between different tools and resources [2]
- development of Web applications (eLearning, ...)

Introduction and Motivation

Solution

A **ReSTful Web Service** for accessing all of these technologies simply by using a browser

Web servicization of LRT allow :

- easy access for non-experts through a Web 2.0 interface
- interoperability between different tools and resources [2]
- development of Web applications (eLearning, ...)



Plan

- 1 Integrated Tools and Resources
 - The Fips Parser
- 2 Characteristics of the WS
 - Architecture of the Web Service
 - Data formats
 - Access to resources
- 3 A Web application based on the WS
 - FipsColor : an eLearning application based on the Service



Plan

- 1 Integrated Tools and Resources
 - The Fips Parser
- 2 Characteristics of the WS
 - Architecture of the Web Service
 - Data formats
 - Access to resources
- 3 A Web application based on the WS
 - FipsColor : an eLearning application based on the Service



What is Fips ?

- A robust & efficient parser for several languages (English, French, German, Italian, Spanish, Greek, ...)
- A flexible object-oriented based platform for multilingual parsing

Some basic properties of Fips

- bottom up parse, with a right-corner attachment procedure
- parallel treatment of alternatives
- use of heuristics to limit the number of alternatives



What is Fips ?

- A robust & efficient parser for several languages (English, French, German, Italian, Spanish, Greek, ...)
- A flexible object-oriented based platform for multilingual parsing

Some basic properties of Fips

- bottom up parse, with a right-corner attachment procedure
- parallel treatment of alternatives
- use of heuristics to limit the number of alternatives



What is Fips ?

- A robust & efficient parser for several languages (English, French, German, Italian, Spanish, Greek, ...)
- A flexible object-oriented based platform for multilingual parsing

Some basic properties of Fips

- bottom up parse, with a right-corner attachment procedure
- parallel treatment of alternatives
- use of heuristics to limit the number of alternatives



What is Fips ?

- A robust & efficient parser for several languages (English, French, German, Italian, Spanish, Greek, ...)
- A flexible object-oriented based platform for multilingual parsing

Some basic properties of Fips

- bottom up parse, with a right-corner attachment procedure
- parallel treatment of alternatives
- use of heuristics to limit the number of alternatives



Grammar

Adaptation of Chomsky's generative grammar

- Structures have all the following schema

$$[_{XP} L X R]$$

where X is the head of the projection XP, L and R are lists of maximal projections (left, right constituents of XP)

- Possible values for X : Adverb, Adjective, Noun, Determiner, Verb, Tense, Preposition, Complementizer, Interjection



Grammar

Adaptation of Chomsky's generative grammar

- Structures have all the following schema

$[_{xp} L X R]$

where **X is the head of the projection XP**, L and R are lists of maximal projections (left, right constituents of XP)

- Possible values for X : Adverb, Adjective, Noun, Determiner, Verb, Tense, Preposition, Complementizer, Interjection



Grammar

Adaptation of Chomsky's generative grammar

- Structures have all the following schema

$[_{XP} \text{ L X R }]$

where X is the head of the projection XP, **L and R are lists of maximal projections** (left, right constituents of XP)

- Possible values for X : Adverb, Adjective, Noun, Determiner, Verb, Tense, Preposition, Complementizer, Interjection



Grammar

Adaptation of Chomsky's generative grammar

- Structures have all the following schema

$$[_{XP} L X R]$$

where X is the head of the projection XP, L and R are lists of maximal projections (left, right constituents of XP)

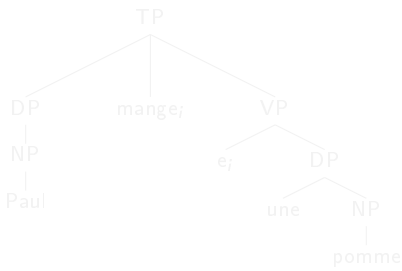
- Possible values for X : Adverb, Adjective, Noun, Determiner, Verb, Tense, Preposition, Complementizer, Interjection



A parsing example

[TP [DP [NP Paul]] mange_i [VP e_i [DP une [NP pomme]]]]

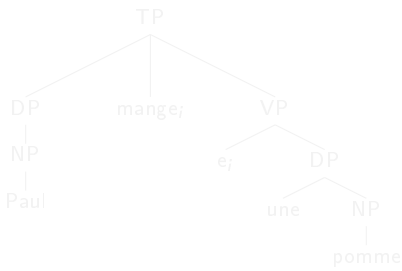
- reads "Paul" and metaprojects a DP structure
- reads "mange" and metaprojects a TP-VP structure. Merge operation with the preceding DP structure
- reads the determiner "une" and creates a DP structure. Merge with the left-adjacent TP constituent, with DP attached as right constituent of the internal VP node
- reads the noun "pomme" and creates an NP structure. Attach it (merge) as a right constituent of the DP structure



A parsing example

[TP [DP [NP Paul]] mange_i [VP e_i [DP une [NP pomme]]]]

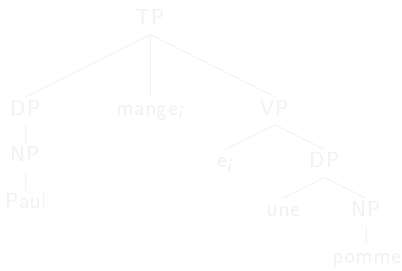
- reads "Paul" and metaprojects a DP structure
- reads "mange" and metaprojects a TP-VP structure. Merge operation with the preceding DP structure
- reads the determiner "une" and creates a DP structure. Merge with the left-adjacent TP constituent, with DP attached as right constituent of the internal VP node
- reads the noun "pomme" and creates an NP structure. Attach it (merge) as a right constituent of the DP structure



A parsing example

[TP [DP [NP Paul]] mange_i [VP e_i [DP une [NP pomme]]]]

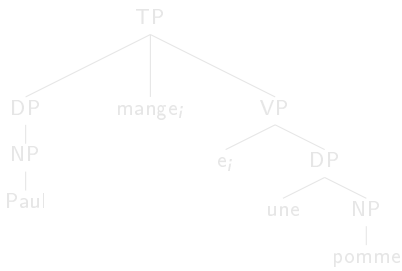
- reads "Paul" and metaprojects a DP structure
- reads "mange" and metaprojects a TP-VP structure. Merge operation with the preceding DP structure
- reads the determiner "une" and creates a DP structure. Merge with the left-adjacent TP constituent, with DP attached as right constituent of the internal VP node
- reads the noun "pomme" and creates an NP structure. Attach it (merge) as a right constituent of the DP structure



A parsing example

[TP [DP [NP Paul]] mange_i [VP e_i [DP une [NP pomme]]]]

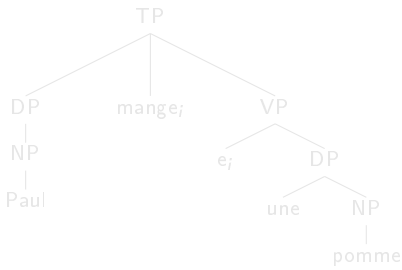
- reads "Paul" and metaprojects a DP structure
- reads "mange" and metaprojects a TP-VP structure. Merge operation with the preceding DP structure
- reads the determiner "une" and creates a DP structure. Merge with the left-adjacent TP constituent, with DP attached as right constituent of the internal VP node
- reads the noun "pomme" and creates an NP structure. Attach it (merge) as a right constituent of the DP structure



A parsing example

[TP [DP [NP Paul]] mange_i [VP e_i [DP une [NP pomme]]]]

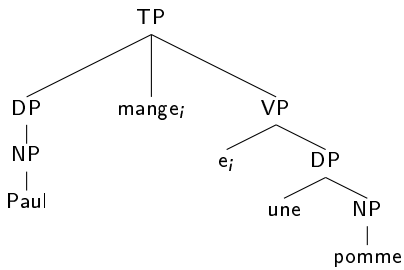
- reads "Paul" and metaprojects a DP structure
- reads "mange" and metaprojects a TP-VP structure. Merge operation with the preceding DP structure
- reads the determiner "une" and creates a DP structure. Merge with the left-adjacent TP constituent, with DP attached as right constituent of the internal VP node
- reads the noun "pomme" and creates an NP structure. Attach it (merge) as a right constituent of the DP structure



A parsing example

[TP [DP [NP Paul]] mange_i [VP e_i [DP une [NP pomme]]]]

- reads "Paul" and metaprojects a DP structure
- reads "mange" and metaprojects a TP-VP structure. Merge operation with the preceding DP structure
- reads the determiner "une" and creates a DP structure. Merge with the left-adjacent TP constituent, with DP attached as right constituent of the internal VP node
- reads the noun "pomme" and creates an NP structure. Attach it (merge) as a right constituent of the DP structure



Evaluation

- Comparative evaluation with the European Parliament corpus
- Parsing of approximately 1 million of words in each language (300000 for Greek)

Language	German	English	Spanish	French	Greek	Italian
Number of symbols	1106559	1075246	1228240	1350522	343461	1181785
Unknown words	10685	5852	9165	5643	6788	9006
Number of Sentences	47058	41488	46216	45694	13328	44124
% of complete analyses	66,54%	75,68%	71,4%	75,97%	51%	70,76%
Speed (word/second)	17	38	132	83	196	112



Plan

- 1 Integrated Tools and Resources
 - The Fips Parser
- 2 Characteristics of the WS
 - Architecture of the Web Service
 - Data formats
 - Access to resources
- 3 A Web application based on the WS
 - FipsColor : an eLearning application based on the Service



A ReSTful Web Service

Services are implemented as "Representational State Transfer" :

- A style of software architecture for distributed hypermedia systems such as the World Wide Web

Why ReST ?

- ReST is simple, both conceptually and programmatically
- ReST only deals with existing well-known standards (HTTP, XML, URI, ...)
- SOAP approach uses full-blown remote objects with remote method invocation and encapsulated functionality



A ReSTful Web Service

Services are implemented as "Representational State Transfer" :

- A style of software architecture for distributed hypermedia systems such as the World Wide Web

Why ReST ?

- ReST is simple, both conceptually and programmatically
- ReST only deals with existing well-known standards (HTTP, XML, URI, ...)
- SOAP approach uses full-blown remote objects with remote method invocation and encapsulated functionality



A ReSTful Web Service

Services are implemented as "Representational State Transfer" :

- A style of software architecture for distributed hypermedia systems such as the World Wide Web

Why ReST ?

- ReST is simple, both conceptually and programmatically
- ReST only deals with existing well-known standards (HTTP, XML, URI, ...)
- **SOAP approach uses full-blown remote objects with remote method invocation and encapsulated functionality**



A ReSTful Web Service

The ReSTful Approach

- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE).
- the identification of resources via URI :
- the operation are stateless :
- the use of standards like HTML, XML or JSON.



A ReSTful Web Service

The ReSTful Approach

- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE);
- the identification of resources via URI :
- the operation are stateless :
- the use of standards like HTML, XML or JSON.



A ReSTful Web Service

The ReSTful Approach

- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE);
- the identification of resources via URI : each resource can be uniquely identified and addressed;
- the operation are stateless :
- the use of standards like HTML, XML or JSON.



A ReSTful Web Service

The ReSTful Approach

- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE);
- the identification of resources via URI : each resource can be uniquely identified and addressed;
- the operation are stateless :
- the use of standards like HTML, XML or JSON.



A ReSTful Web Service

The ReSTful Approach

- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE);
- the identification of resources via URI : each resource can be uniquely identified and addressed;
- the operation are stateless : the client is responsible for sending all the information required for a successful method execution;
- the use of standards like HTML, XML or JSON.



A ReSTful Web Service

The ReSTful Approach

- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE);
- the identification of resources via URI : each resource can be uniquely identified and addressed;
- the operation are stateless : the client is responsible for sending all the information required for a successful method execution;
- the use of standards like HTML, XML or JSON.



A ReSTful Web Service

The ReSTful Approach

- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE);
- the identification of resources via URI : each resource can be uniquely identified and addressed;
- the operation are stateless : the client is responsible for sending all the information required for a successful method execution;
- the use of standards like HTML, XML or JSON.



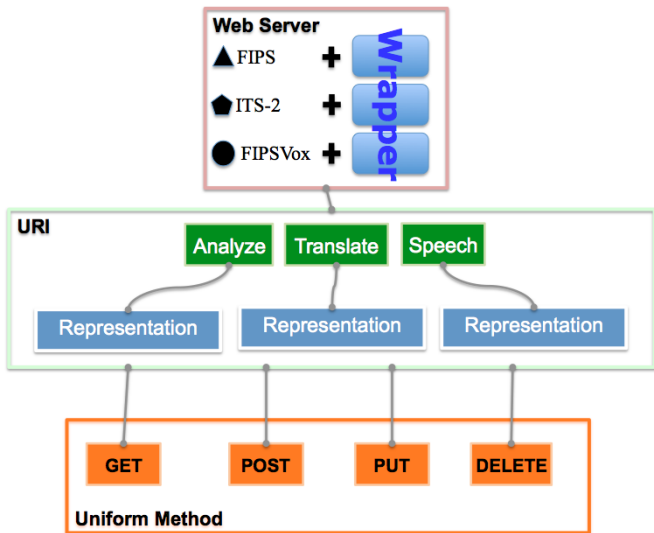
A ReSTful Web Service

The ReSTful Approach

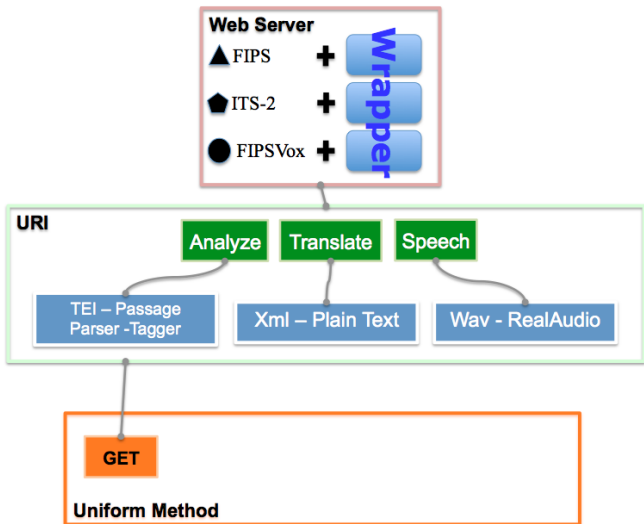
- the use of the "Uniform Interface" : resources can be manipulated using HTTP protocol and the method (PUT, GET, HEAD, POST, DELETE);
- the identification of resources via URI : each resource can be uniquely identified and addressed;
- the operation are stateless : the client is responsible for sending all the information required for a successful method execution;
- the use of standards like HTML, XML or JSON.



A ReSTful Web Service



A ReSTful Web Service



TEI : Text Encoding Initiative

```

<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <s xml:lang="french">
    <phr type="DP" function="SUBJ">
      <w type="DETERMINANT-DEFINI SIN MAS" lemma="le">le</w>
      <w type="NOM-COMMUN SIN MAS" lemma="chat">chat</w>
    </phr>
    <phr type="" function="Predicate">
      <w type="VERBE-IND-PRE 3 SIN" lemma="manger">mange</w>
    </phr>
    <phr type="DP" function="OBJ">
      <w type="DETERMINANT-DEFINI SIN FEM" lemma="le">la</w>
      <w type="NOM-COMMUN SIN FEM" lemma="souris">souris</w>
    </phr>
  </s>
</TEI>

```

- Tokenization - Lemmatization
- PoS Tagging
- Chunking



TEI : Text Encoding Initiative

```

<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <s xml:lang="french">
    <phr type="DP" function="SUBJ">
      <w type="DETERMINANT-DEFINI SIN MAS" lemma="le">le</w>
      <w type="NOM-COMMUN SIN MAS" lemma="chat">chat</w>
    </phr>
    <phr type="" function="Predicate">
      <w type="VERBE-IND-PRE 3 SIN" lemma="manger">mange</w>
    </phr>
    <phr type="DP" function="OBJ">
      <w type="DETERMINANT-DEFINI SIN FEM" lemma="le">la</w>
      <w type="NOM-COMMUN SIN FEM" lemma="souris">souris</w>
    </phr>
  </s>
</TEI>

```

- Tokenization - Lemmatization
- PoS Tagging
- Chunking



TEI : Text Encoding Initiative

```

<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <s xml:lang="french">
    <phr type="DP" function="SUBJ">
      <w type="DETERMINANT-DEFINI SIN MAS" lemma="le">le</w>
      <w type="NOM-COMMUN SIN MAS" lemma="chat">chat</w>
    </phr>
    <phr type="" function="Predicate">
      <w type="VERBE-IND-PRE 3 SIN" lemma="manger">mange</w>
    </phr>
    <phr type="DP" function="OBJ">
      <w type="DETERMINANT-DEFINI SIN FEM" lemma="le">la</w>
      <w type="NOM-COMMUN SIN FEM" lemma="souris">souris</w>
    </phr>
  </s>
</TEI>

```

- Tokenization - Lemmatization
- PoS Tagging
- Chunking



TEI : Text Encoding Initiative

```

<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <s xml:lang="french">
    <phr type="DP" function="SUBJ">
      <w type="DETERMINANT-DEFINI SIN MAS" lemma="le">le</w>
      <w type="NOM-COMMUN SIN MAS" lemma="chat">chat</w>
    </phr>
    <phr type="" function="Predicate">
      <w type="VERBE-IND-PRE 3 SIN" lemma="manger">mange</w>
    </phr>
    <phr type="DP" function="OBJ">
      <w type="DETERMINANT-DEFINI SIN FEM" lemma="le">la</w>
      <w type="NOM-COMMUN SIN FEM" lemma="souris">souris</w>
    </phr>
  </s>
</TEI>

```

- Tokenization - Lemmatization
- PoS Tagging
- **Chunking**



PASSAGE

```

<Sentence id="E1">
  <T id="E1W0T0" start="1" end="5">Paul</T>
  <T id="E1W1T0" start="6" end="10">eats</T>
  <T id="E1W2T0" start="11" end="13">an</T>
  <T id="E1W3T0" start="14" end="19">apple</T>
  <G id="E1G0" type="GN">
    <W id="E1W0" tokens="E1W0T0" pos="properNoun" lemma="Paul" />
  </G>
  <G id="E1G1" type="NV">
    <W id="E1W1" tokens="E1W1T0" pos="verb" lemma="eat" />
  </G>
  <G id="E1G2" type="GN">
    <W id="E1W2" tokens="E1W2T0" pos="indefiniteDeterminer" lemma="a" />
    <W id="E1W3" tokens="E1W3T0" pos="commonNoun" lemma="apple" />
  </G>
  <R id="E1R0" type="SUJ-V">
    < sujet ref="E1W0T0" />
    < verbe ref="E1W1T0" />
  </R>
  <R id="E1R1" type="COD-V">
    < cod ref="E1W3T0" />
    < verbe ref="E1W1T0" />
  </R>
</Sentence>

```

- Tokenization
- Lemmatization
- PoS Tagging
- Chunking
- Syntactic Relation



PASSAGE

```

<Sentence id="E1">
  <T id="E1W0T0" start="1" end="5">Paul</T>
  <T id="E1W1T0" start="6" end="10">eats</T>
  <T id="E1W2T0" start="11" end="13">an</T>
  <T id="E1W3T0" start="14" end="19">apple</T>
  <G id="E1G0" type="GN">
    <W id="E1W0" tokens="E1W0T0" pos="properNoun" lemma="Paul" />
  </G>
  <G id="E1G1" type="NV">
    <W id="E1W1" tokens="E1W1T0" pos="verb" lemma="eat" />
  </G>
  <G id="E1G2" type="GN">
    <W id="E1W2" tokens="E1W2T0" pos="indefiniteDeterminer" lemma="a" />
    <W id="E1W3" tokens="E1W3T0" pos="commonNoun" lemma="apple" />
  </G>
  <R id="E1R0" type="SUJ-V">
    < sujet ref="E1W0T0" />
    < verbe ref="E1W1T0" />
  </R>
  <R id="E1R1" type="COD-V">
    < cod ref="E1W3T0" />
    < verbe ref="E1W1T0" />
  </R>
</Sentence>

```

- Tokenization
- Lemmatization
- PoS Tagging
- Chunking
- Syntactic Relation



PASSAGE

```

<Sentence id="E1">
  <T id="E1W0T0" start="1" end="5">Paul</T>
  <T id="E1W1T0" start="6" end="10">eats</T>
  <T id="E1W2T0" start="11" end="13">an</T>
  <T id="E1W3T0" start="14" end="19">apple</T>
  <G id="E1G0" type="GN">
    <W id="E1W0" tokens="E1W0T0" pos="properNoun" lemma="Paul" />
  </G>
  <G id="E1G1" type="NV">
    <W id="E1W1" tokens="E1W1T0" pos="verb" lemma="eat" />
  </G>
  <G id="E1G2" type="GN">
    <W id="E1W2" tokens="E1W2T0" pos="indefiniteDeterminer" lemma="a" />
    <W id="E1W3" tokens="E1W3T0" pos="commonNoun" lemma="apple" />
  </G>
  <R id="E1R0" type="SUJ-V">
    <sujet ref="E1W0T0" />
    <verbe ref="E1W1T0" />
  </R>
  <R id="E1R1" type="COD-V">
    <cod ref="E1W3T0" />
    <verbe ref="E1W1T0" />
  </R>
</Sentence>

```

- Tokenization
- Lemmatization
- PoS Tagging
- Chunking
- Syntactic Relation



PASSAGE

```

<Sentence id="E1">
  <T id="E1W0T0" start="1" end="5">Paul</T>
  <T id="E1W1T0" start="6" end="10">eats</T>
  <T id="E1W2T0" start="11" end="13">an</T>
  <T id="E1W3T0" start="14" end="19">apple</T>
  <G id="E1G0" type="GN">
    <W id="E1W0" tokens="E1W0T0" pos="properNoun" lemma="Paul" />
  </G>
  <G id="E1G1" type="NV">
    <W id="E1W1" tokens="E1W1T0" pos="verb" lemma="eat" />
  </G>
  <G id="E1G2" type="GN">
    <W id="E1W2" tokens="E1W2T0" pos="indefiniteDeterminer" lemma="a" />
    <W id="E1W3" tokens="E1W3T0" pos="commonNoun" lemma="apple" />
  </G>
  <R id="E1R0" type="SUJ-V">
    < sujet ref="E1W0T0" />
    < verbe ref="E1W1T0" />
  </R>
  <R id="E1R1" type="COD-V">
    < cod ref="E1W3T0" />
    < verbe ref="E1W1T0" />
  </R>
</Sentence>

```

- Tokenization
- Lemmatization
- PoS Tagging
- Chunking
- Syntactic Relation



PASSAGE

```

<Sentence id="E1">
<T id="E1W0T0" start="1" end="5">Paul</T>
<T id="E1W1T0" start="6" end="10">eats</T>
<T id="E1W2T0" start="11" end="13">an</T>
<T id="E1W3T0" start="14" end="19">apple</T>
<G id="E1G0" type="GN">
  <W id="E1W0" tokens="E1W0T0" pos="properNoun" lemma="Paul" />
</G>
<G id="E1G1" type="NV">
  <W id="E1W1" tokens="E1W1T0" pos="verb" lemma="eat" />
</G>
<G id="E1G2" type="GN">
  <W id="E1W2" tokens="E1W2T0" pos="indefiniteDeterminer" lemma="a" />
  <W id="E1W3" tokens="E1W3T0" pos="commonNoun" lemma="apple" />
</G>
<R id="E1R0" type="SUJ-V">
  < sujet ref="E1W0T0" />
  < verbe ref="E1W1T0" />
</R>
<R id="E1R1" type="COD-V">
  < cod ref="E1W3T0" />
  < verbe ref="E1W1T0" />
</R>
</Sentence>

```

- Tokenization
- Lemmatization
- PoS Tagging
- **Chunking**
- Syntactic Relation



PASSAGE

```

<Sentence id="E1">
<T id="E1W0T0" start="1" end="5">Paul</T>
<T id="E1W1T0" start="6" end="10">eats</T>
<T id="E1W2T0" start="11" end="13">an</T>
<T id="E1W3T0" start="14" end="19">apple</T>
<G id="E1G0" type="GN">
  <W id="E1W0" tokens="E1W0T0" pos="properNoun" lemma="Paul" />
</G>
<G id="E1G1" type="NV">
  <W id="E1W1" tokens="E1W1T0" pos="verb" lemma="eat" />
</G>
<G id="E1G2" type="GN">
  <W id="E1W2" tokens="E1W2T0" pos="indefiniteDeterminer" lemma="a" />
  <W id="E1W3" tokens="E1W3T0" pos="commonNoun" lemma="apple" />
</G>
<R id="E1R0" type="SUJ-V">
  < sujet ref="E1W0T0" />
  < verbe ref="E1W1T0" />
</R>
<R id="E1R1" type="COD-V">
  < cod ref="E1W3T0" />
  < verbe ref="E1W1T0" />
</R>
</Sentence>

```

- Tokenization
- Lemmatization
- PoS Tagging
- Chunking
- Syntactic Relation



PASSAGE

```

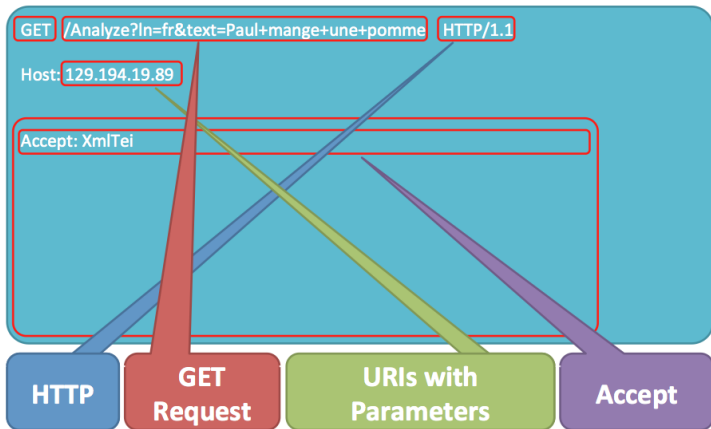
<Sentence id="E1">
  <T id="E1W0T0" start="1" end="5">Paul</T>
  <T id="E1W1T0" start="6" end="10">eats</T>
  <T id="E1W2T0" start="11" end="13">an</T>
  <T id="E1W3T0" start="14" end="19">apple</T>
  <G id="E1G0" type="GN">
    <W id="E1W0" tokens="E1W0T0" pos="properNoun" lemma="Paul" />
  </G>
  <G id="E1G1" type="NV">
    <W id="E1W1" tokens="E1W1T0" pos="verb" lemma="eat" />
  </G>
  <G id="E1G2" type="GN">
    <W id="E1W2" tokens="E1W2T0" pos="indefiniteDeterminer" lemma="a" />
    <W id="E1W3" tokens="E1W3T0" pos="commonNoun" lemma="apple" />
  </G>
  <R id="E1R0" type="SUJ-V">
    < sujet ref="E1W0T0" />
    < verbe ref="E1W1T0" />
  </R>
  <R id="E1R1" type="COD-V">
    < cod ref="E1W3T0" />
    < verbe ref="E1W1T0" />
  </R>
</Sentence>

```

- Tokenization
- Lemmatization
- PoS Tagging
- Chunking
- Syntactic Relation



Example of Request



The Response

HTTP/1.1 200 OK

Date: Fri, 16 Sep 2011 09:02:25 GMT
 Content-length : 572
 Content-language : fr
 Content-type: application/xml; charset=utf-8

```
<xml version="1.0" encoding="UTF-8">
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <phr type="DP" function="SUBJ">
    <w type="NOM-PROPRE SIN MAS" lemma="Paul">Paul</w>
  </phr>
  <!-- etc -->
</TEI>
```

HTTP Error Codes

Body



Plan

- 1 Integrated Tools and Resources
 - The Fips Parser
- 2 Characteristics of the WS
 - Architecture of the Web Service
 - Data formats
 - Access to resources
- 3 A Web application based on the WS
 - FipsColor : an eLearning application based on the Service

FipsColor

Analyse syntaxique couleur pour
l'apprentissage du français

Plus d'informations

Jean est parti dans l'est de la France.

Analyser

Jean est parti dans l'est de la France .

VERBE-AUX-IND 3 SIN -

lemme : être

NOM-COMMUN SIN MAS -

lemme : est

A+ A-A

Catégories lexicales

Nom Verbe Adjectif Préposition Déterminant Adverbe Conjonction Pronom Tout/Annuler

Fonctions grammaticales

Sujet Prédicat Objet Direct Objet Indirect Attribut Circonstanciel Tout/Annuler



UNIVERSITÉ
DE GENÈVE

LABORATORY OF ANALYSIS
AND TECHNOLOGY OF LANGUAGE

Conclusion - Further Work

- WS increase the accessibility and usability of multilingual and multi-format LRT
- Services can be useful not only for researchers in linguistics
- In a future version :
 - Add other languages : Romanian, Japanese and Portuguese
 - Inclusion of Semantic Web technology to provide search and reasoning

References

- [1] J-P. Goldman, A. Gaudinat, L. Nerima, and E. Wehri.
Fipsvox : a french tts based on a syntactic parser.
In Speech Synthesis Workshop, Edinburgh, 2001.

- [2] Marta Villegas, Núria Bel, Santiago Bel, and Víctor Rodríguez.
A case study on interoperability for language resources and applications.
In Proceedings of the LREC'10, Valletta, Malta, 2010.

- [3] E. Wehri.
Fips, a deep linguistic multilingual parser.
In ACL 2007 Workshop on deep Linguistic Processing, pages 120–127, Prague, Czech Republic, 2007.

- [4] E. Wehri and L. Nerima.
Traduction multilingue : le projet multa.
In JEP – TALN – RECITAL Conference, Avignon, France, 2008.



THANK YOU