Matxin: Moving towards language independence

Aingeru Mayor, Francis Tyers

IXA Taldea
Euskal Herriko Unibertsitatea
aingeru@ehu.es
.
Dept. Lleng. i Sist. Informtics,
Universitat d'Alacant
ftyers@prompsit.com

2009.eko urriaren 29

Introduction

- *Matxin* RBMT system
  - Developed by the IXA group
  - First publicly available MT system for translating into Basque
  - Initial aim: translate into Basque
  - Architecture designed to be independent of both source and target languages
- *Matxin* 1.0, es→eu
  - Open-source version available for free download (GPL), with a reduced bilingual lexicon:
    `http://matxin.sourceforge.net`
  - Full version available for testing online:
    `http://www.opentrad.org`

Introduction

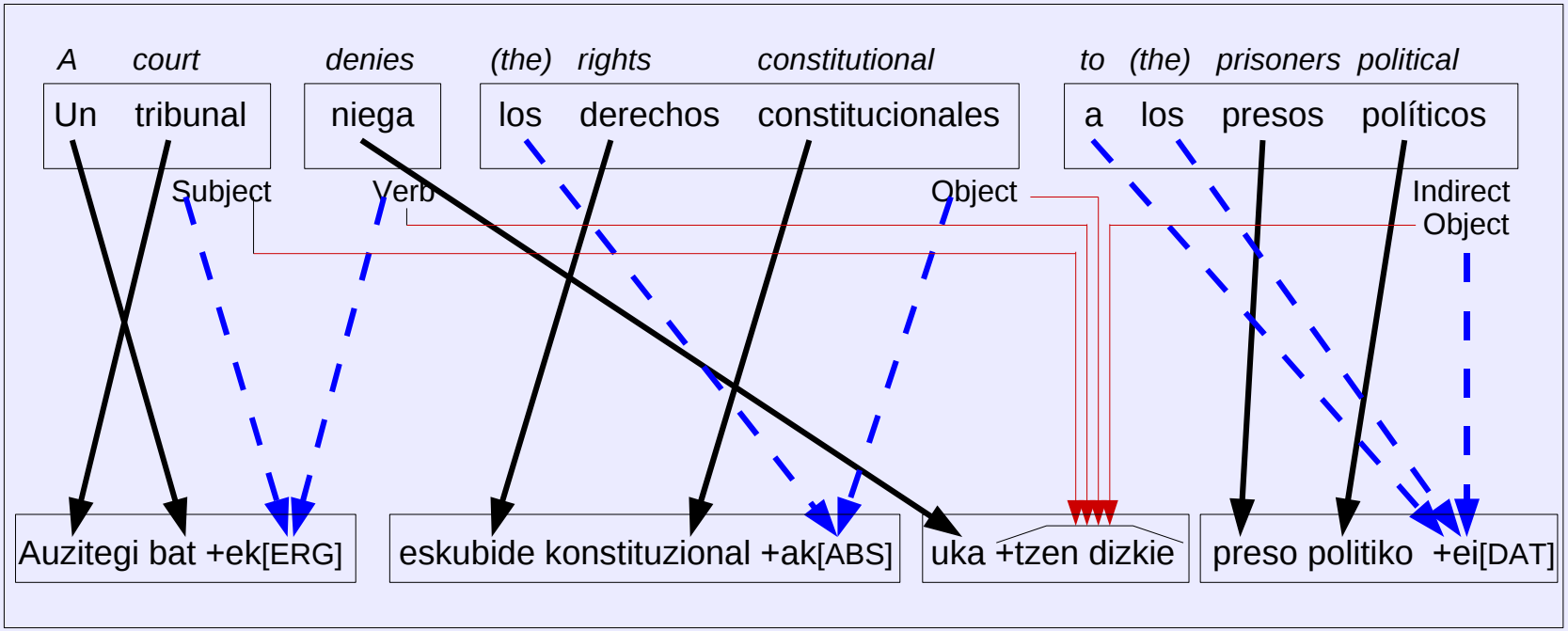
- The system is being adapted to new langauge pairs
  - Development team: en→eu
  - Independent initiative: br→en
- The experience of adapting to other languages
  - reveals problems in the current architecture and implementation
  - gives ideas of how to improve the system with respect to its language independence

Index

Index

Translating from Spanish to Basque, a complex task

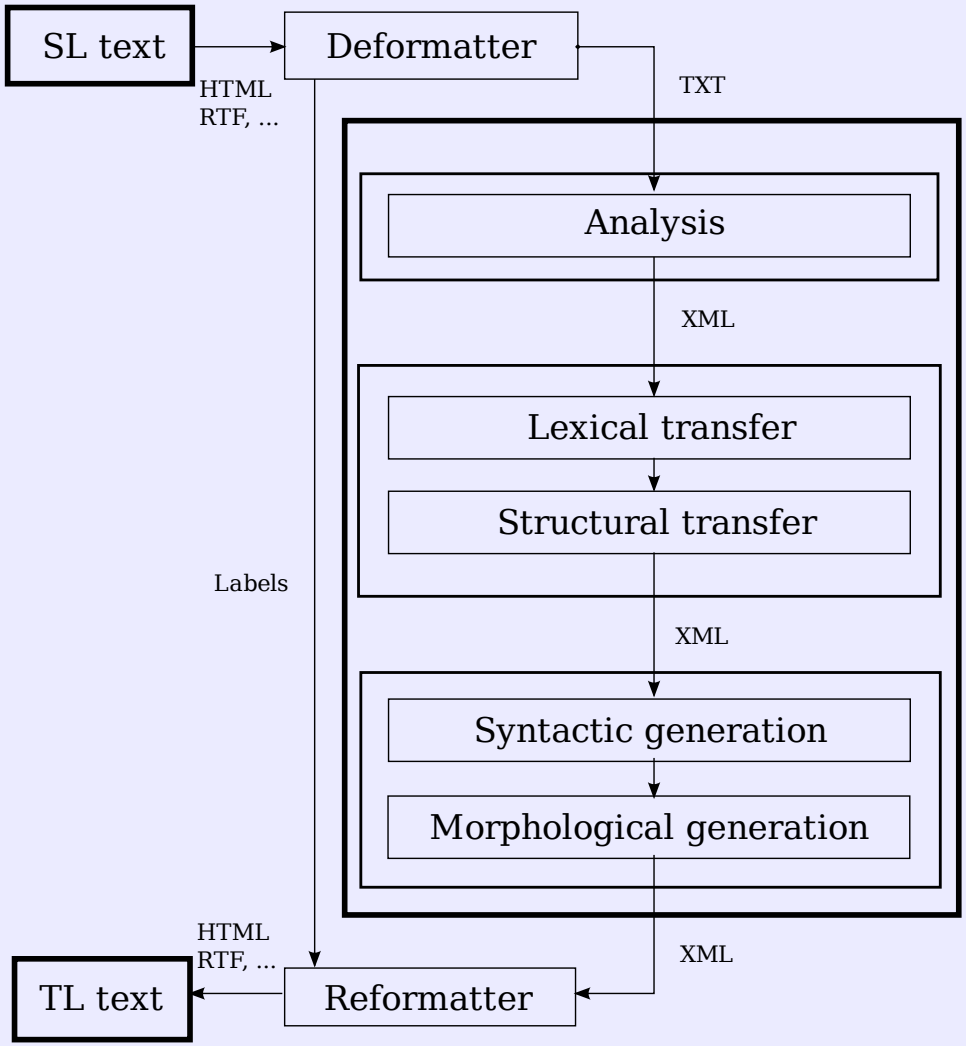
Un tribunal niega los derechos constitucionales a los presos políticos



Characteristics

- RBMT
  - Difficulties for SMT
    - Morphologically-rich language
    - Limited digital resources
- Classic transfer-based model
  - analysis, transfer and generation
- Design guided by the translation linguistic tasks
- Linguistic data separated from algorithms
- Monolingual modules as independent as possible from bilingual modules

Architecture



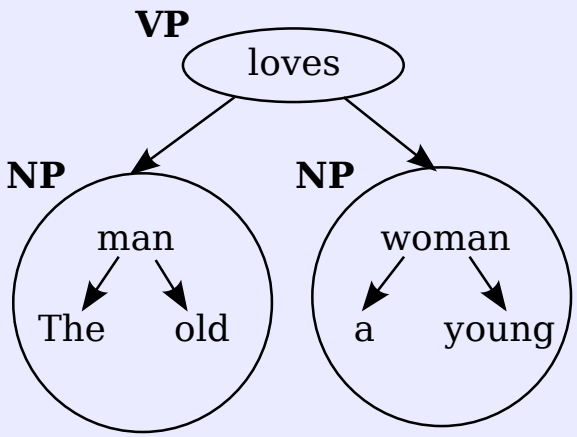
Reusability and use of standards

- We have reused previously developed
  - modules: Spanish analyser, morphological Basque generator, de-formatter and re-formatter, lexical transfer, etc.
  - linguistic resources: dictionaries and corpus
- The modules and the linguistic data created could be reused
  - Spanish dependency analyser, dictionary of prepositions, verbal chunk transfer, etc.
- Consequences of reuse: resource and module heterogeneity
- To ensure interoperability: necessary to use standards
  - Dictionaries: coded in a format based on XML according to the *Apertium* specification
  - Translation data structure: based on XML

Translation data structure

- Processed by the transfer and generation modules
- Used for the communication between modules

- Based in a hybrid syntactic structure
  - the constituents are labelled
  - the dependency relations are expressed:
    - between the words of each of the constituents
    - between the constituents



Translation data structure. DTD

- The DTD describes
  - main elements of the translation process
    - sentences: the basic translation unit
    - chunks: broadly equivalent to a constituent
    - nodes: a word or a multiword term
  - attributes
    - linguistic information and also document format
  - dependency relations
    - one element containing another element, indicates that it comes below in its dependency structure

Translation data structure. DTD

```
<!ELEMENT SENTENCE (CHUNK+)>
<!ATTLIST SENTENCE
    ord    CDATA     <!--Order in the whole text-->
    ref    CDATA     <!--Corresponding SL sentence-->
    alloc  CDATA     <!--Position of the 1st character-->
>
<!ELEMENT CHUNK (NODE, CHUNK*)>
<!ATTLIST CHUNK
    ord    CDATA     <!--Order in the sentence-->
    ref    CDATA     <!--Corresponding SL chunk-->
    alloc  CDATA     <!--Position of the 1st character-->
    type   CDATA     <!--Chunk type-->
    si     CDATA     <!--Syntactic information-->
    focus  CDATA     <!--Focus-->
    prep   CDATA     <!--Preposition-->
    trans  CDATA     <!--Transitivity-->
    subper CDATA     <!--Subject's person-->
    <!...>
>
```

Translation data structure. DTD

```
<!ELEMENT NODE (NODE*)>
<!ATTLIST NODE
    ord     CDATA     <!--Order in the chunk-->
    ref     CDATA     <!--Corresponding SL node-->
    alloc   CDATA     <!--Position of the 1st character-->
    form    CDATA     <!--Form-->
    lem     CDATA     <!--Lemma-->
    mi      CDATA     <!--Morphological information-->
    pos     CDATA     <!--Part-of-speech-->
    suf     CDATA     <!--Information on the suffix-->
    det     CDATA     <!--Determination-->
    num     CDATA     <!--Number-->
    per     CDATA     <!--Person-->
    loc     CDATA     <!--Location-information-->
    <!...>
>
```

Analysis

- FreeLing:
  - Developed at the UPC
  - Open-source
  - Morphological analysis, part-of-speech tagging, partial-parsing and dependency analysis

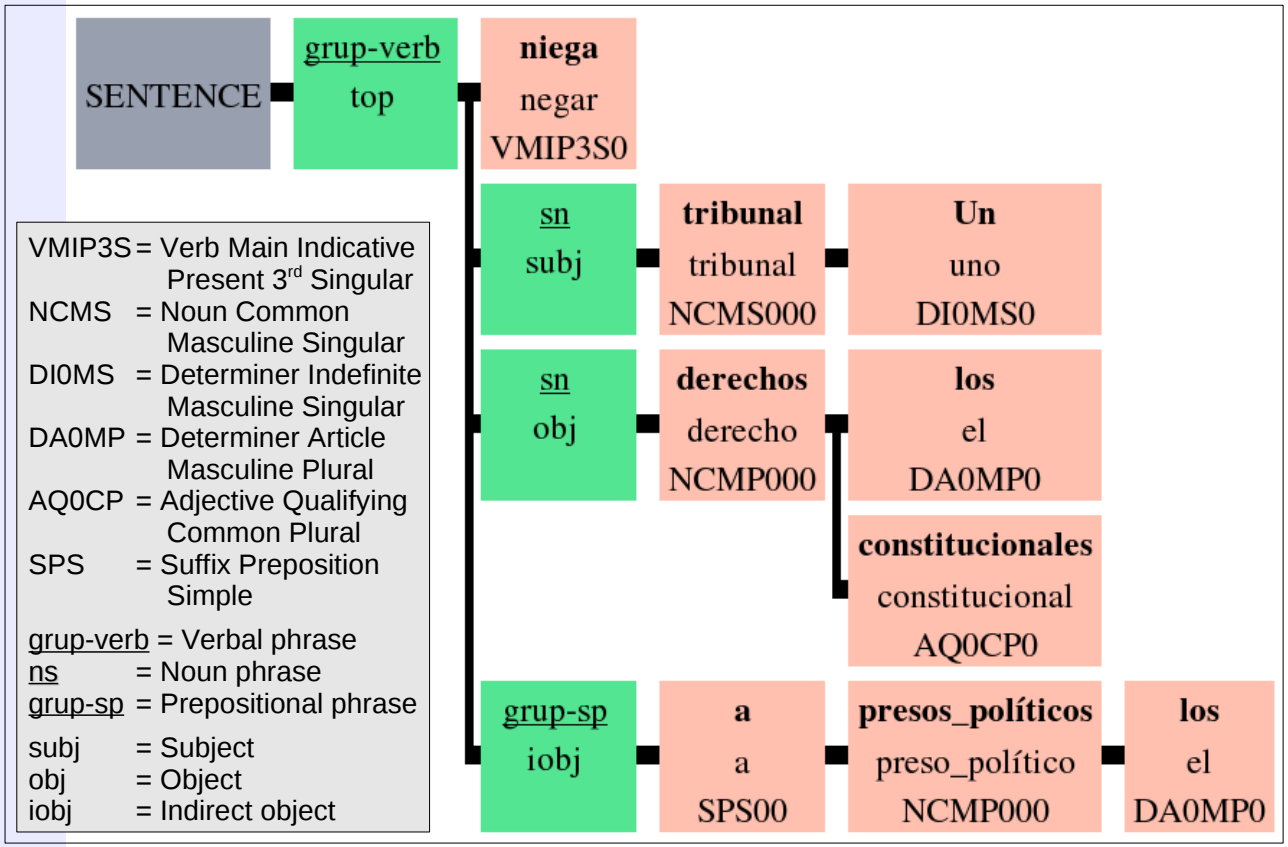
Output of the analysis

Un tribunal niega los derechos constitucionales a los presos políticos

```
<SENTENCE ord='1' alloc='0'>
  <CHUNK ord='2' alloc='12' type='grup-verb' si='top'>
    <NODE ord='1' alloc='12' form='niega' lem='negar' mi='VMIP3S0'>
    <CHUNK ord='1' alloc='0' type='sn' si='subj'>
      <NODE ord='2' alloc='3' form='tribunal' lem='tribunal' mi='NCMS000'>
        <NODE ord='1' alloc='0' form='Un' lem='uno' mi='DI0MS0'/>
      </NODE>
    </CHUNK>
    <CHUNK ord='3' alloc='18' type='sn' si='obj' focus='true'>
      <NODE ord='2' alloc='22' form='derechos' lem='derecho' mi='NCMP000'>
        <NODE ord='1' alloc='18' form='los' lem='el' mi='DA0MP0'/>
        <NODE ord='3' alloc='31' form='constitucionales' lem='constitucional' mi='AQ0CP0'/>
      </NODE>
    </CHUNK>
    <CHUNK ord='4' alloc='48' type='grup-sp' si='iobj'>
      <NODE ord='1' alloc='48' form='a' lem='a' mi='SPS00'>
        <NODE ord='3' alloc='54' form='presos_politicos' lem='preso_politico' mi='NCMP000'>
          <NODE ord='2' alloc='50' form='los' lem='el' mi='DA0MP0'/>
        </NODE>
      </NODE>
    </CHUNK>
  </CHUNK>
</SENTENCE>
```

Output of the analysis

Un tribunal niega los derechos constitucionales a los presos políticos

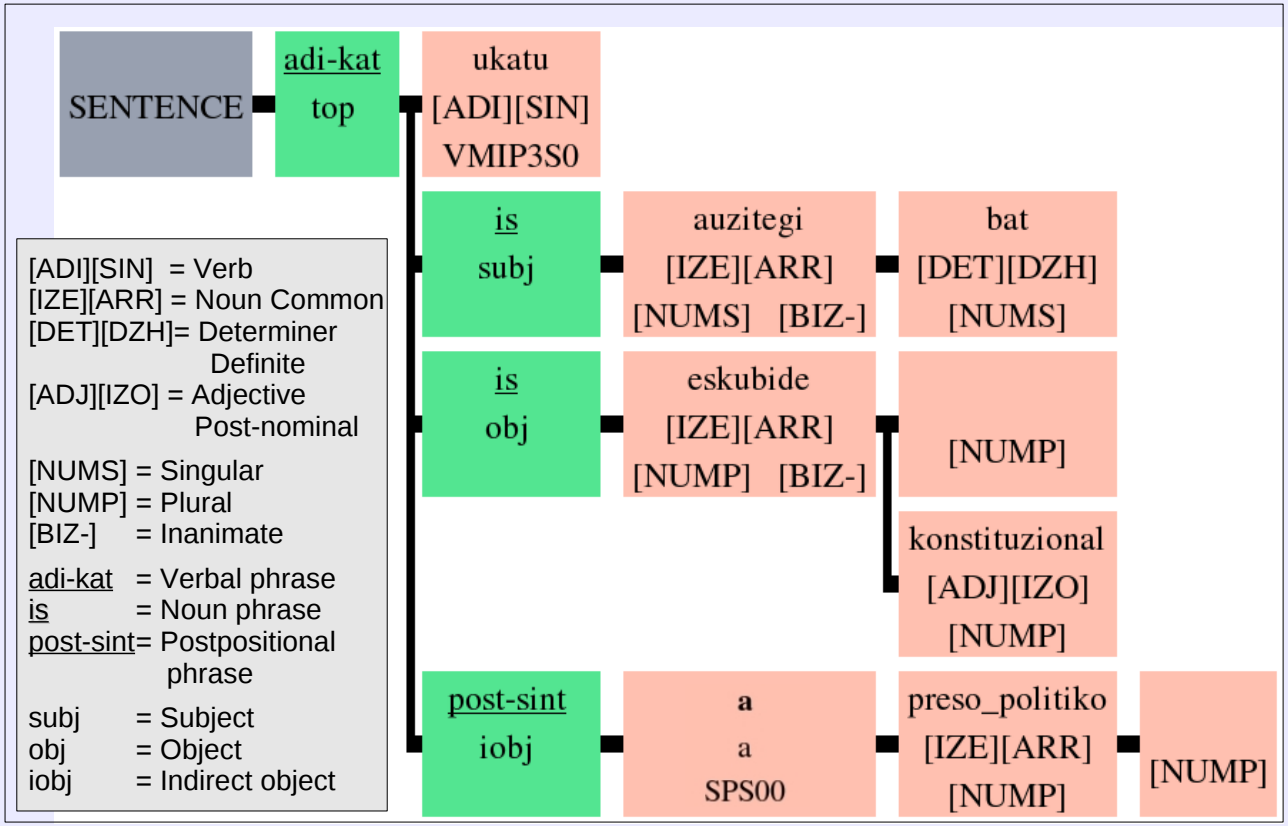


Lexical transfer

- Search in the lexicon
  - Input: source node's lemma and morphological information
  - Output: target equivalent's lemma, part-of-speech, location information, person, number, morphological composition and other features
- In some cases lexical transfer is not required:
  - Nodes containing prepositions
  - Nodes corresponding to verbal chunks which are not the root

Output of the lexical transfer

Un tribunal niega los derechos constitucionales a los presos políticos

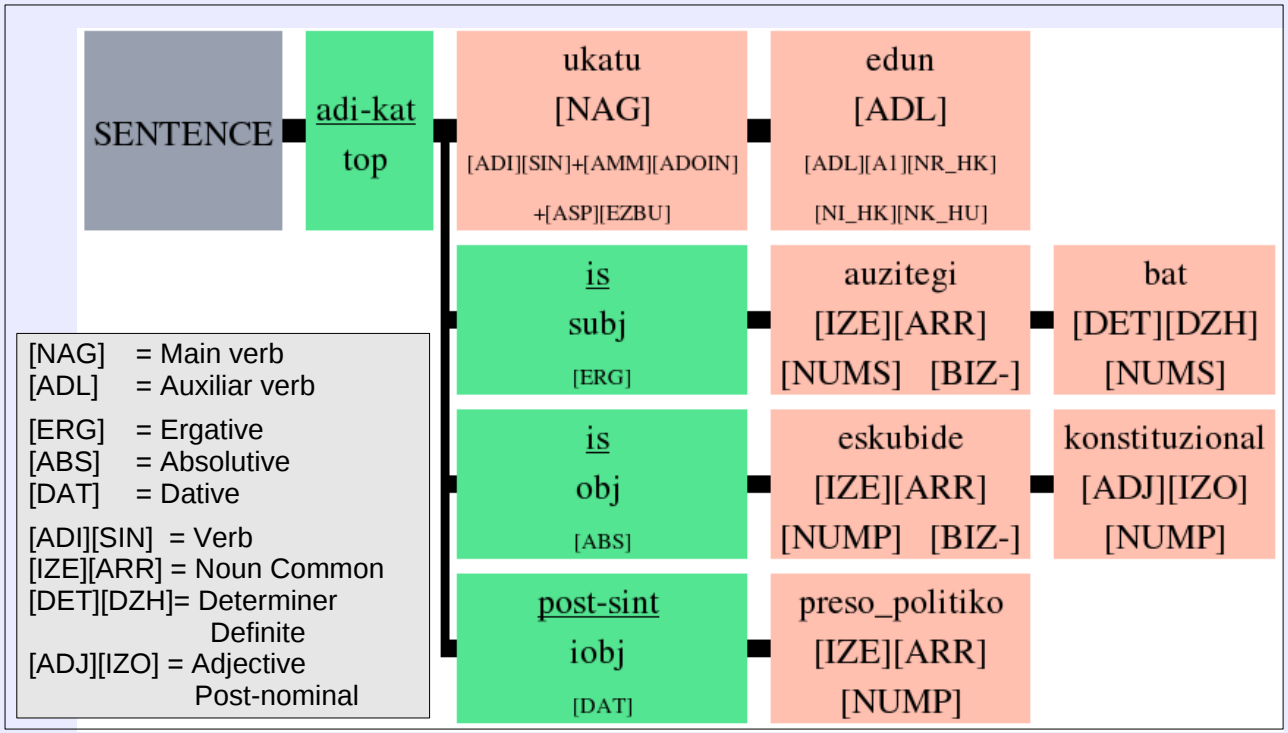


Structural transfer

1. Operations inside the chunk:
   1. Moving information from node to chunk
   2. Deleting nodes not containing lexical information
2. Transfer of prepositions and syntactic functions
3. Operations between chunks:
   1. Determining the person attribute for the subject and direct object
   2. Moving information from chunk to chunk
   3. Deleting chunks without nodes
4. Transfer of verbal chunks
5. Adaptation operations

Output of the structural transfer

Un tribunal niega los derechos constitucionales a los presos políticos



Syntactic generation

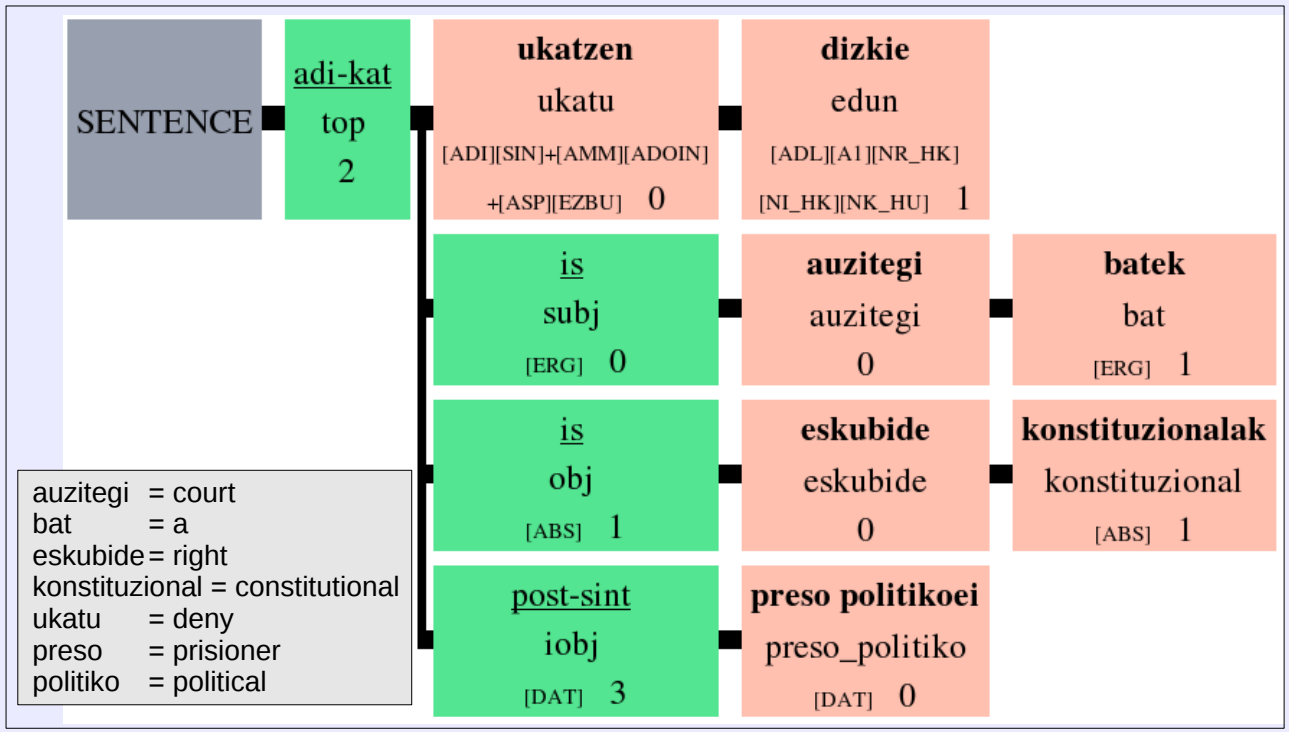
1. Ordering the nodes within the chunk
   - A precedence rule has been coded for each type of chunk
2. In chunks with postpositional information (generally not verbal chunks), this information is added to the last node in the chunk
3. Ordering the chunks within the sentence
   1. Determining the relative order for each pair of parent-child chunks
   2. Determining the absolute order of the chunks in the sentence

Morphological generation

- Only words including morphological information for generation will be processed
  - In verbal chunks: all the nodes
  - In the rest: only the last node
- *Morfeus*, the morphological processor for Basque created by the IXA group

Output of the generation

Un tribunal niega los derechos constitucionales a los presos políticos



0		1		2		3
Auzitegi batek		eskubide konstituzionalak		ukatzen dizkie		preso politikoei
0	1	0	1	0	1	0

Evaluation

- Human-targeted Translation Edit Rate, HTER
  - HTER = 40%
- Not yet suitable for unrestricted use in text dissemination.

Evaluation

- Evaluation in the framework of the virtual expert *AnHitz*
  - 30%: 'very good', 'good' or 'quite good'
  - 39%: 'comprehensible'.
- Useful for content assimilation (for understanding a text)

Index

Freeling: pros...

- FreeLing is fairly straightforward to add a new language
- It works well for languages with low or medium inflection written in the western Latin alphabet (English, Spanish)
- Language data available for
  - (with chunking and dependency parsing): Asturian, Catalan, English and Spanish
  - (without it): Welsh, Galician, Italian and Portuguese

Freeling: ...and cons

- Problems with:
  - Morphologically complex languages (Basque, Sámi) or languages with productive compounding (Icelandic, Norwegian)
    - The morphology of a language is described by way of a full-form list
    - When each word can have many inflected forms, the size of the full-form list becomes unmanageable
  - Characters outside of `latin1` (Welsh)

Alternatives

- Morphological analysis
  - Free toolkits for implementing finite-state morphologies
    - appropriate for complex morphologies: `foma`, `hfst`
    - not appropriate for complex morphologies: `lttoolbox`
- Dependecy analysis
  - Desirable to make it possible the use of Constraint Grammar
  - VISL Constraint Grammar
    - Freely-available rule-based parsers available (Faroese, North Sámi)

Rules for handling the translation data structure

- Each module of the transfer and generation phases
  - takes as input the translation data structure
  - walks its elements
  - applying a set of rules or running some operations on them

  - Some specific tasks are done by external modules:
    - Search in the bilingual lexicon
    - Translation of prepositions and syntactic functions
    - Translation of verbal chunks
    - Morphological generation

- The rules for each of the modules have different formats
  - Most of them are implemented as tab-delimited files

Examples of the current rules

- Rule for interchunk movements (Structural transfer)

OriginChunk		DestinationChunk			
Condition	/Attrib.	Condition	/Attrib.	direction	writeMode
`si='subj'`	`/per`	`type='verb-ch'`	`/subjPer`	`up`	`overwrite`

- Rule for interchunk ordering (Syntactic generation)

parentChunkType(x1)	childChunkType(x2)	condition	order
`verb-chain`	`.*?`	`focus=true`	`x2.x1`

Other problems

- A few linguistic decisions are expressed in the code
  - To append semantic information to nouns, the tag for noun `[IZE]` is hard-coded in the lexical transfer module
  - The deletion of nodes that do not have any lexical information, and chunks that do not have any node is coded directly in the modules of the structural transfer
- No validation mechanism for the rules

- All of these characteristics make it difficult to modify the rules or to extend the system to new language pairs

Proposed rule formalism

- We have designed a single XML-based format for all the rules
- Existing rules (and decisions coded in the modules) have been converted by hand (65 rules)
- The interpreter that will apply the rules has not been yet developed

Proposed rule formalism: DTD

- Each rule is made up of two parts
  - Pattern match
  - Set of actions
- DTD:

```
<!ELEMENT rule (match, actions)>
<!ATTLIST rule id CDATA #REQUIRED>
<!ELEMENT match (def+)>
<!ELEMENT def>
<!ELEMENT actions (act+)>
<!ELEMENT act>
```

Proposed rule formalism

- Pattern match
  - Configuration to be searched for in the data structure
  - Elements defined by XPath based expressions
- Set of actions
  - Action(s) to apply to the elements defined in the pattern match
  - Include main operations
    - assignment, deletion, substitution and concatenation
  - and calls to external functions
    - searching in the lexicon, using the morphological processor...
- The interpreter will
  - evaluate the XPath expressions defined in the pattern match
  - collect the references of the elements,
  - apply the actions specified in the rules to them.

Proposed rule formalism: Examples (I)

```
<!--- Copy person-information from subject to verb-chunk>
<rule id='1'>
  <match>
    <def> C1 := //CHUNK[@type='verb-chunk']</def>
    <def> C2 := ./CHUNK[@si='subj'] </def>
  </match>
  <actions>
    <act> C1/@subjper := C2/@per </act>
  </actions>
</rule>

<!--- Order verb and focus chunk>
<rule id='2'>
  <match>
    <def> C1 := //CHUNK[@type='verb-chunk']</def>
    <def> C2 := ./CHUNK[@focus='true'] </def>
  </match>
  <actions>
    <act> C2/@relord := 'left-jointly' </act>
  </actions>
</rule>
```

Proposed rule formalism: Examples (II)

```
<!--- Delete nodes without lexical value>
<rule id='3'>
  <match>
    <def> N := //NODE[not(@lem)] </def>
  </match>
  <actions>
    <act> delete(N) </act>
  </actions>
</rule>

<!--- Search semantic information for nouns>
<rule id='4'>
  <match>
    <def> N := //NODE[@pos='[IZE][ARR]] </def>
  </match>
  <actions>
    <act> N/@sem := &semDict(N/@lem) </act>
  </actions>
</rule>
```

Proposed rule formalism

- It guarantees that all the linguistic information is coded in declarative rules
- It makes much more easier
  - to add or modify the rules
  - to create new sets of rules for new language pairs

Conclusion

- The experience of adapting *Matxin* to new language pairs has revealed problems
- We have proposed some solutions
  - More flexible source language analysis
  - A unified rule formalism for handling the translation data structure in the transfer and generation phases
- These solutions will make much more easier to modify the system and to adapt it for new language pairs

Matxin: Moving towards language independence

Aingeru Mayor, Francis Tyers

IXA Taldea
Euskal Herriko Unibertsitatea
aingeru@ehu.es
.
Dept. Lleng. i Sist. Informtics,
Universitat d'Alacant
ftyers@prompsit.com

2009.eko urriaren 29