

# An FST grammar for verb chain transfer in a Spanish-Basque MT System

Iñaki Alegria, Arantza Díaz de Ilarraza, Gorka Labaka, Mikel Lersundi, Aingeru Mayor and Kepa Sarasola

IXA Research Group on NLP  
http://ixa.si.ehu.es



Faculty of Computer Science  
University of the Basque Country

## Context

- Spanish-Basque MT system:
  - Traditional transfer model
  - Based on shallow and dependency parsing
- Integrated in OpenTrad initiative:
  - MT engines for translation among main languages in Spain
  - Open, reusable and interoperable framework
  - Government-funded and shared among different universities and companies

## Task

- Structural transfer of verb chains
- Complex due to the high distance between both languages
- Corpus coverage: 92%
  - non-finite forms (21%), indicative forms (65%), periphrases (6%)

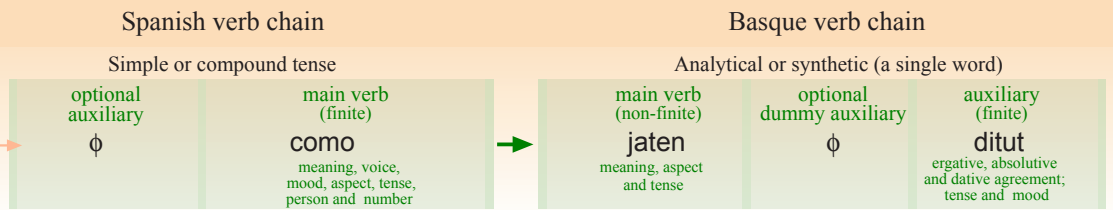
## Implementation

- FST grammar
- Using XRCE Finite State Tools

## Verb chain transfer

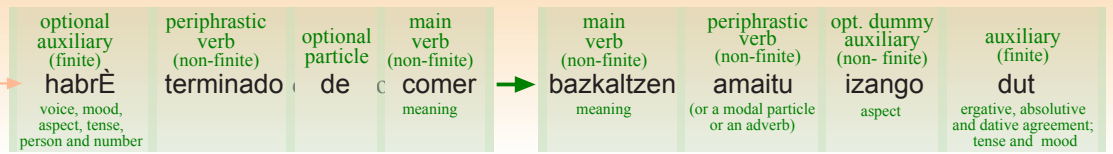
### Finite verbs

Yo como manzanas  
Nik sagarrak jaten ditut  
I eat apples



### Periphrases

Yo habré terminado de comer  
Nik bazkaltzen amaitu izango dut  
I will have finished eating



## FST Grammar

### Input

- Morphological information of the nodes of the Spanish verb chain
- Basque form corresponding to the Spanish main verb of the chain, and information about its transitivity
- Agreement information about the objects (absolutive and dative) and the type of subordination of the sentence

### Rules

- Spanish verb type identification and Basque schema adding rules  
[ esVerbChainType @-> ... "=" euVerbChainSchema ]

Add one of the possible Basque verb chain schema depending on the type of the Spanish verb chain: non-finite, non-periphrastic verbs and four types of periphrastic verbs **6 rules**

- Attribute replacement rules  
[ "euAttr" @-> "euVal" || ?\* esVals ?\* "=" ?\* euVals ?\* \_ ]

Replace attributes in the schema with their corresponding values, depending on the values of some attributes in the Spanish verb chain and/or in the Basque schema **80 rules**

- Cleaning rules

Remove the unnecessary information **2 rules**

### Output

- The list of the nodes of the Basque verb chain, each one with the information necessary to
  - decide the order of the words (between parenthesis)
  - carry out the morphological generation (between brackets)

## An example

Input *porque no habré tenido que comer patatas* (because I won't have had to eat potatoes)  
haber[vaif1s] + tener[vmpp] + que[cs] + comer[vmn] // jan [trans] // [o3p] [caus]

### FST

- A rule identifies the input as a Spanish periphrastic verb chain of type 1 and adds the schema for the Basque verb for this type  
[ esVerbChainTypePerif1 @->... "=" euVerbChainSchemaP1 ]

haber[vaif1s]+tener[vmpp]+que[cs]+comer[vmn] // jan [trans] // [obj3p] [caus]  
=>P1> (main) Aspm / Per Aspp / Dum Aspd / Aux TenseM Erg Abs + RelM

- Other rules replace one by one the attributes of the Basque verb schema
- |   |
|---|
| [iPeri @-> ibehar(per)i    ?* iteneri ?* iquei ?* "=" iPl1 ?* _ ] |
| [iAspp1 @-> i[partPerf]i    ?* VAIF ?* "=" iPl1 ?* _ ]            |
| [iAuxi @-> iedun(aux)i    ?* "=" ?* ibehar(per)i ?* _ ]           |
| [iErgi @-> i[sls]i    ?* ils1 ?* i=>1 ?* iedun(aux)i ?* _ ]       |

=>P1> (main) [partPerf] / behar(per) [partPerf] / izan(dum) [partFut] / edun(aux) [indPres] [sls] [o3p] +lako[causMorph]

- The last rules eliminate the information of the input

### Output

jan (main) [partPerf] / behar (per) [partPerf] / izan (dum) [partFut] / edun (aux) [indPres] [sls] [o3p] +lako [causMorph]

ez ditudalako patatak jan behar izango