Developing an open-source FST grammar for verb chain transfer in a Spanish-Basque MT System

Aingeru Mayor, Mans Hulden, Gorka Labaka
Ixa Group
University of the Basque Country
aingeru@ehu.es, mhulden@email.arizona.edu, gorka.labaka@ehu.es

Abstract

This paper presents the current status of development of a finite state transducer grammar for the verbal-chain transfer module in *Matxin*, a Rule Based Machine Translation system between Spanish and Basque. Due to the distance between Spanish and Basque, the verbal-chain transfer is a very complex module in the overall system. The grammar is compiled with *foma*, an open-source finite-state toolkit, and yields a translation execution time of 2000 verb chains/second.

1 Introduction

This paper presents the current status of development of an FST (Finite State Transducer) grammar we have developed for *Matxin*, a Machine Translation system between Spanish and Basque.

Basque is a minority language isolate, and it is likely that an early form of this language was already present in Western Europe before the arrival of the Indo-European languages.

Basque is a highly inflected language with free order of sentence constituents. It is an agglutinative language, with a rich flexional morphology.

Basque is also a so-called ergative-absolutive language where the subjects of intransitive verbs appear in the absolutive case (which is unmarked), and where the same case is used for the direct object of a transitive verb. The subject of the transitive verb (that is, the agent) is marked differently, with the ergative case (in Basque by the suffix *-k*). The presence of this morpheme also triggers main and auxiliary verbal agreement. Auxiliary verbs, or 'periphrastic' verbs, which accompany most main verbs, agree not only with the subject, but also with the direct object and the indirect object, if present. Among European languages, this polypersonal system (multiple verb agreement) is rare, and found only in Basque, some Caucasian languages, and Hungarian.

The fact that Basque is both a morphologically rich and less-resourced language makes the use of statistical approaches for Machine Translation difficult and raises the need to develop a rule-based architecture which in the future could be combined with statistical techniques.

The *Matxin es-eu* (Spanish-Basque) MT engine is a classic transfer-based system comprising three main modules: analysis of the Spanish text (based on *FreeLing*, (Atserias et al., 2006)), transfer, and generation of the Basque target text.

In the transfer process, lexical transfer is first carried out using a bilingual dictionary coded in the XML format of Apertium dictionary files (.dix) (Forcada et al., 2009), and compiled, using the FST library implemented in the Apertium project (the *lt-toolbox* library), into a finite-state transducer that can be processed very quickly.

Following this, structural transfer at the sentence level is performed, and some information is transferred from some chunks[1] to others while some chunks may be deleted. Finally, the structural trans-

[1] A chunk is a non-recursive phrase (noun phrase, prepositional phrase, verbal chain, etc.) which expresses a constituent (Abney, 1991; Civit, 2003). In our system, chunks play a crucial part in simplifying the translation process, due to the fact that each module works only at a single level, either inside or between chunks.

Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing, pages 65–69,
Donostia–San Sebastián, July 23–25, 2012. ©2012 Association for Computational Linguistics

fer at the verb chunk level is carried out. The verbal chunk transfer is a very complex module because of the nature of Spanish and Basque auxiliary verb constructions, and is the main subject of this paper.

This verb chain transfer module is implemented as a series of ordered replacement rules (Beesley and Karttunen, 2003) using the *foma* finite-state toolkit (Hulden, 2009). In total, the system consists of 166 separate replacement rules that together perform the verb chunk translation. In practice, the input is given to the first transducer, after which its output is passed to the second, and so forth, in a cascade. Each rule in the system is unambiguous in its output; that is, for each input in a particular step along the verb chain transfer, the transducers never produce multiple outputs (i.e. the transducers in question are functional). Some of the rules are joined together with composition, yielding a total of 55 separate transducers. In principle, all the rules could be composed together into one monolithic transducer, but in practice the size of the composed transducer is too large to be feasible. The choice to combine some transducers while leaving others separate is largely a memory/translation speed tradeoff.

2 Spanish and Basque verb features and their translation

In the following, we will illustrate some of the main issues in translating Spanish verb chains to Basque. Since both languages make frequent use of auxiliary verb constructions, and since periphrastic verb constructions are frequent in Basque, transfer rules can get quite complex in their design.

For example, in translating the phrase

(Yo)	compro	(una	manzana)
(I)	buy	(an	apple)
[PP1CSN00]	**[VMIP1S0]**	**[DI0FS0]**	**[NCFS000]**

we can translate it using the imperfective participle form (*erosten*) of the verb *erosi (to buy)*, and a transitive auxiliary (*dut*) which itself contains both subject agreement information (*I*: 1st sg.) and number agreement with the object (*an apple*: 3rd sg.): *(nik) (sagar bat) erosten dut*. The participle carries information concerning meaning, aspect and tense, whereas the auxiliaries convey information about argument structure, tense and mood.

Table 1 illustrates the central idea of the verb chunk transfer. In the first four examples the form of the transitive auxiliary changes to express agreement with different ergative arguments (the subject of the clause), absolutive arguments (the direct object) and dative arguments (the indirect object). In the fifth example the future participle is used. The last example shows the translation of a periphrastic construction, in which the the Spanish and the Basque word orders are completely different: this is reflected in the Spanish *tengo que*-construction (have to) which appears before the main verb, whereas in the Basque, the equivalent (*behar*) appears after the main verb (*erosi*).

3 The FST grammar

We carry out the verbal chunk transfer using finite-state transducers (Alegria et al., 2005). The grammar rules take as input the Spanish verbal chunk, perform a number of transformations on the input, and then create and output the verbal chunk for Basque.

To illustrate the functioning of the grammar, let us consider the following example sentence in Spanish:

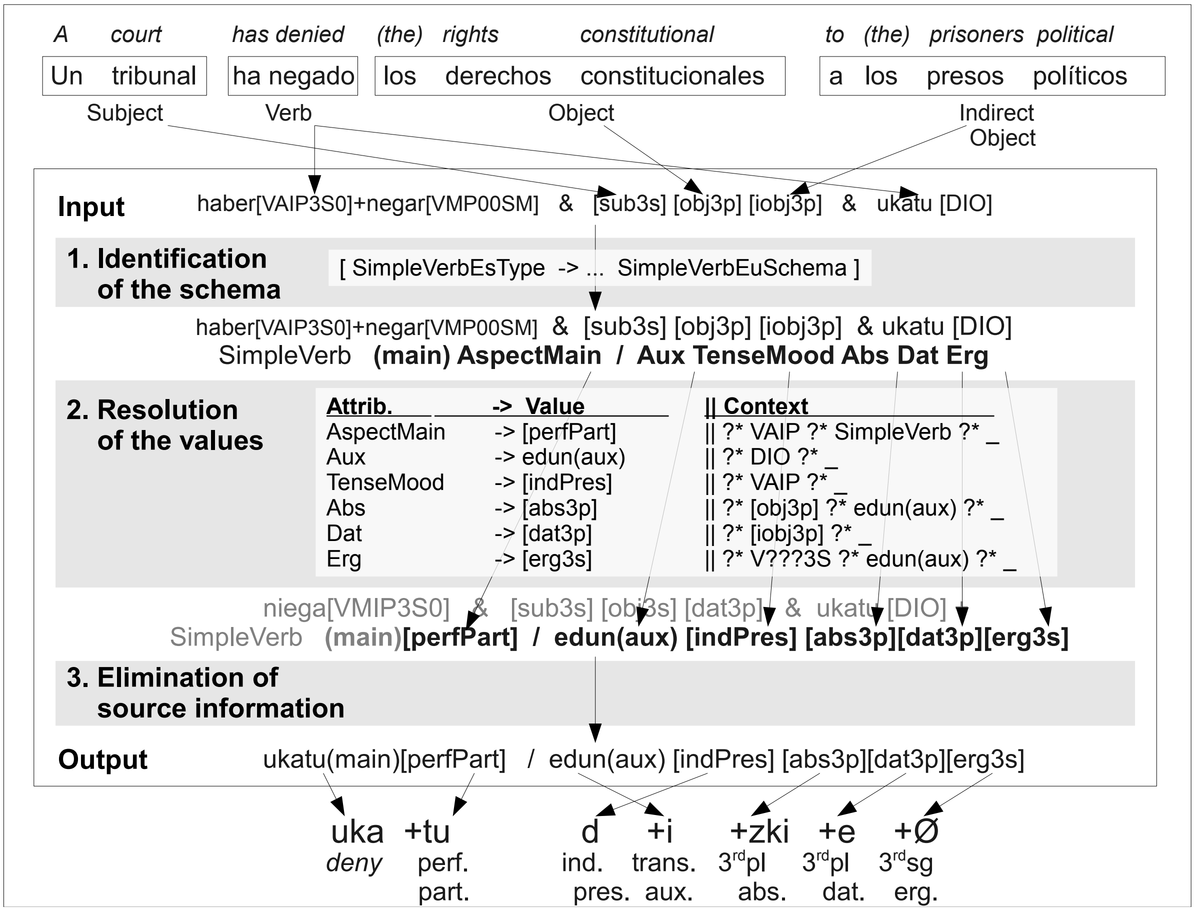
"*Un tribunal ha negado los derechos constitucionales a los presos polticos*" (A court has denied constitutional rights to political prisoners). The correct translation into Basque given by the system for this example is as follows: *Auzitegi batek eskubide konstituzionalak ukatu dizkie preso politikoei*. Figure 1 shows a detailed overview of how the whole transfer of the verbal chunk is performed for this particular example.

First, the input to the grammar is assumed to be a string containing (separated by the '&' symbol) the following information :

- the morphological information (using EAGLES-style tags Leech and Wilson (1996)) for all nodes (separated by '+' symbol) in the Spanish verbal chunk (**haber[VAIP3S0]+negar[VMP00SM]**);
- the morphological information of the subject (**[sub3s]**), the direct object (**[obj3p]**) and the indirect object (**[iobj3p]**);
- the translation of the main verb in Basque (**ukatu**) and information about its transitivity

Spanish sentence	English	Basque translation
(Yo) compro (una manzana)	(I) buy (an apple)	(Nik) (sagar bat) erosten dut
(Yo) compro (manzana**s**)	(I) buy (apple**s**)	(Nik) (sagarra**k**) erosten d**it**ut
(**Tú**) compr**as** (manzanas)	(**You**) buy (apples)	(**Zuk**) (sagarrak) erosten ditu**zu**
(Yo) (**te**) compro (una manzana)	(I) buy (**you**) (an apple)	(Nik) (**zuri**) (sagar bat) erosten d**izu**t
(Yo) compr**aré** (una manzana)	(I) **will** buy (an apple)	(Nik) (sagar bat) erosi**ko** dut
(Yo) **tengo que** comprar (manzanas)	(I) **must** buy (apples)	(Nik) (sagarrak) erosi **behar** ditut

Table 1: Examples of translations



A court has denied (the) rights constitutional to (the) prisoners political
Un tribunal
ha negado
los derechos constitucionales
a los presos políticos
Subject
Verb
Object
Indirect Object
Input
haber[VAIP3S0]+negar[VMP00SM] & [sub3s] [obj3p] [iobj3p] & ukatu [DIO]
1. Identification of the schema
[SimpleVerbEsType -> ... SimpleVerbEuSchema]
haber[VAIP3S0]+negar[VMP00SM] & [sub3s] [obj3p] [iobj3p] & ukatu [DIO]
SimpleVerb (main) AspectMain / Aux TenseMood Abs Dat Erg
2. Resolution of the values
Attrib. -> Value || Context
AspectMain -> [perfPart] || ?* VAIP ?* SimpleVerb ?* _
Aux -> edun(aux) || ?* DIO ?* _
TenseMood -> [indPres] || ?* VAIP ?* _
Abs -> [abs3p] || ?* [obj3p] ?* edun(aux) ?* _
Dat -> [dat3p] || ?* [iobj3p] ?* _
Erg -> [erg3s] || ?* V???3S ?* edun(aux) ?* _
niega[VMIP3S0] & [sub3s] [obj3s] [dat3p] & ukatu [DIO]
SimpleVerb (main)[perfPart] / edun(aux) [indPres] [abs3p][dat3p][erg3s]
3. Elimination of source information
Output
ukatu(main)[perfPart] / edun(aux) [indPres] [abs3p][dat3p][erg3s]
uka +tu d +i +zki +e +Ø
deny perf. part. ind. pres. trans. aux. 3rd pl abs. 3rd pl dat. 3rd sg erg.

Figure 1: Example of the transfer of a verbal chunk.

([**DIO**]), indicating a ditransitive construction:

```
  haber[VAIP3S0]+negar[VMP00SM] &
[sub3s][obj3p][iobj3p] & ukatu[DIO]
```

The grammatical rules are organized into three groups according to the three main steps defined for translating verbal chunks:

1. Identification of the Basque verbal chunk schema corresponding to the source verbal chunk.

   There are twelve rules which perform this task, each of which corresponds to one of the following verbal chunks in Spanish: non-conjugated verbs, simple non-periphrastic verbs as well as four different groups reserved for the periphrastic verbs.

   The verbal chunk of the example in figure 1 is a simple non-periphrastic one, and the rule that handles this particular case is as follows:

   ```
   [simpleVerbEsType -> ...
   simpleVerbEuSchema]
   ```

   When this rule matches the input string representing a simple non-periphrastic verbal chunk (`simpleVerbEsType`) it adds the corresponding Basque verbal chunk schema (`simpleVerbEuSchema`) to the end of the input string. `simpleVerbEsType` is a complex automaton that has the definition of the Spanish simple verbs. `simpleVerbEuSchema` is the type of the verbal chunk (`SimpleVerb`) and an automaton that contains as strings the pattern of elements (separated by the '/' symbol) that the corresponding Basque verb chunk will need to have (in this case, the main verb and the auxiliary verb):

   ```
   SimpleVerb (main) AspectMain /
   Aux TenseMood Abs Dat Erg
   ```

2. Resolution of the values for the attributes in the Basque schema.

   A total of 150 replacement rules of this type have been written in the grammar. Here are some rules that apply to the above example:

   ```
   [AspectMain -> [perfPart] || ?* VAIP
   ?* SimpleVerb ?* _ ]
   ```

   ```
   [Aux -> edun(aux) || ?* DIO ?* _ ]
   ```

   ```
   [Abs -> [abs3p] || ?* [obj3p] ?*
   edun(aux) ?* _ ]
   ```

3. Elimination of source-language information (4 rules in total).

   The output of the grammar for the example is:

   ```
   ukatu(main)[perfPart] /
   edun(aux)[indPres][abs3p][dat3p][erg3s]
   ```

   The first node has the main verb (*ukatu*) with the perfective participle aspect, and the second one contains the auxiliary verb (*edun*) with all its morphological information: indicative present and argument structure.

In the output string, each of the elements contains the information needed by the subsequent syntactic generation and morphological generation phases.

4 Implementation

When the verbal chunk transfer module was first developed, there did not exist any efficient open-source tools for the construction of finite state transducers. At the time, the *XFST*-toolkit (Beesley and Karttunen, 2003) was used to produce the earlier versions of the module: this included 25 separate transducers of moderate size, occupying 2,795 kB in total. The execution speed was roughly 250 verb chains per second. Since *Matxin* was designed to be open source, we built a simple compiler that converted the *XFST* rules into regular expressions that could then be applied without FST technology, at the cost of execution speed. This verbal chunk transfer module read and applied these regular expressions at a speed of 50 verbal chunks per second.

In the work presented here, we have reimplemented and expanded the original rules written for *XFST* with the *foma*[2] toolkit (Hulden, 2009). After adapting the grammar and compiling it, the 55 separate transducers occupy 607 kB and operate at roughly 2,000 complete verb chains per second.[3] Passing the strings from one transducer to the next in the chain of 55 transducers in accomplished by the depth-first-search transducer chaining functionality available in the *foma* API.

[2] http://foma.sourceforge.net

[3] On a 2.8MHz Intel Core 2 Duo.

References

Abney, S. (1991). *Principle-Based Parsing: Computation and Psycholinguistics*, chapter Parsing by Chunks, pages 257–278. Kluwer Academic, Boston.

Alegria, I., Díaz de Ilarraza, A., Labaka, G., Lersundi, M., Mayor, A., and Sarasola, K. (2005). An FST grammar for verb chain transfer in a Spanish–Basque MT system. In *Finite-State Methods and Natural Language Processing*, volume 4002, pages 295–296, Germany. Springer Verlag.

Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., and Padró, M. (2006). Freeling 1.3: Syntactic and semantic services in an open-source NLP library. In *Proceedings of LREC*, volume 6, pages 48–55.

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI Publications, Stanford, CA.

Civit, M. (2003). *Criterios de etiquetación y desambiguación morfosintáctica de corpus en Español*. PhD thesis, Universidad de Barcelona.

Forcada, M., Bonev, B. I., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sanchez, G., Sánchez-Martínez, F., Armentano-Oller, C., Montava, M. A., Tyers, F. M., and Ginestí-Rosell, M. (2009). Documentation of the open-source shallow-transfer machine translation platform Apertium. Technical report, Departament de Llenguatges i Sistemes Informatics. Universitat d'Alacant. Available at http://xixona.dlsi.ua.es/ fran/apertium2-documentation.pdf.

Hulden, M. (2009). Foma: a finite-state compiler and library. In *Proceedings of EACL 2009*, pages 29–32.

Leech, G. and Wilson, A. (1996). EAGLES recommendations for the morphosyntactic annotation of corpora. *Technical report, EAGLES Expert Advisory Group on Language Engineering Standards, Istituto di Linguistica Computazionale, Pisa, Italy*.