An Open Architecture for Transfer-based Machine Translation between Spanish and Basque

Iñaki I. Alegria, Arantza Díaz de Ilarraza, Gorka G. Labaka, Mikel M. Lersundi, Aingeru A. Mayor, Kepa K. Sarasola, Mikel M. Forcada, Sergio S. Ortiz, Lluís L. Padró

An Open Architecture for Transfer-based Machine Translation between Spanish and Basque

Iñaki Alegria[1], Arantza Diaz de Ilarraza[1], Gorka Labaka[1], Mikel Lersundi[1], Aingeru Mayor[1], Kepa Sarasola[1], Mikel L. Forcada[2], Sergio Ortiz-Rojas[2], Lluís Padró[3]

[1]IXA Taldea, Informatika Fakultatea,
Euskal Herriko Unibertsitatea, E-20071 Donostia
[2]Transducens group, Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant
[3]TALP group, Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Abstract

We present the current status of development of an open architecture for the translation from Spanish into Basque. The machine translation architecture uses an open source analyser for Spanish and new modules mainly based on finite-state transducers. The project is integrated in the *OpenTrad* initiative, a larger government-funded project shared among different universities and small companies, which will also include MT engines for translation among the main languages in Spain. The main objective is the construction of an open, reusable and interoperable framework. This paper describes the design of the engine, the formats it uses for the communication among the modules, the modules reused from other project named *Matxin* and the new modules we are building.

1. Introduction

This paper presents the current status of development and the main structure of an open source MT engine which translates from Spanish into Basque using the traditional transfer model and based on shallow and dependency parsing.

The project is based on the previous work of our group (Díaz de Ilarraza et al., 2000) but is now integrated in the *OpenTrad* initiative, a larger government-funded project shared among different universities and small companies (Corbí-Bellot et al., 2005), which will also include MT engines for translation among the main languages in Spain. The main objective of this initiative is the construction of an open, reusable and interoperable framework.

One of the main novelties of this architecture is that it will be released under an open source license (together with pilot linguistic data derived from other open source projects such as *Freeling* or created specially for this purpose) and will be distributed free of charge. This means that anyone having the necessary computational and linguistic skills will be able to adapt or enhance it to produce a new MT system, even for other pairs of related languages or other NLP applications. The whole system will be released at the beginning of 2006.

In the *OpenTrad* project two different but coordinated designs have been carried out. The differences are due to the distance between the languages:

1. An open source shallow-transfer machine translation engine for the Romance languages of Spain (the main ones being Spanish, Catalan and Galician). The MT architecture proposed uses finite-state transducers for lexical processing, hidden Markov models for part-of-speech tagging, and finite-state based chunking for structural transfer, and is largely based upon that of systems already developed by the

Transducens group such as InterNOSTRUM (Spanish-Catalan, Canals-Marote et al., 2001) and Traductor Universia (Spanish-Portuguese, Garrido-Alenda, 2003).

2. A deeper-transfer engine for the Spanish—Basque pair, which will be described in this paper.

Some of the components (modules, data formats and compilers) from the first architecture will also be useful for the second. Indeed, an important additional goal of this work is testing which modules from the first architecture can be integrated in deeper-transfer architectures for more difficult language pairs.

We expect that the introduction of an open source MT architecture will help finding solutions for well known problems in MT systems: having different technologies for different pairs, closed-source architectures being hard to adapt to new uses, etc. It would also help shifting the current business model from a licence-based one to a service-based one, and favour the interchange of existing linguistic data through the use of the XML-based formats defined in the project.

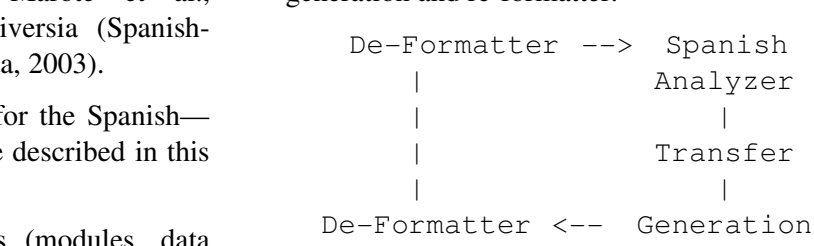
The following sections give an overview of the architecture (sec. 2), the formats defined for the interoperation among the different modules (sec. 3), the encoding of linguistic data (sec. 4), and finally, we give some concluding remarks (sec. 5).

2. The MT architecture

The engine is a classical transfer system consisting of 3 main components: analysis of Spanish, transfer from Spanish to Basque and generation of the Basque output.

It is based on the previous work of our group (Díaz de Ilarraza et al., 2000) but with new features and a new aim: interoperability with other linguistic resources and convergence with the other engines in the *OpenTrad* project through the use of XML . The previous object-oriented architecture is being rewritten into a open source one which will use modules which are shared with other engines in the *OpenTrad* project and will comply with its format specifications.

The main modules are five: de-formatter, Spanish analysis based on *FreeLing* (Carreras et al., 2004), Spanish-Basque transfer, Basque generation and re-formatter.



The following sections describe each module. The transfer and generation phases work in three levels: lexical form (tagged as node), chunk and sentence.

No semantic disambiguation is applied, but a large number of multi-word units representing collocations, named-entities and complex terms are being included in the bilingual dictionary in order to minimize this limitation.

2.1. The de-formatter

The *de-formatter* separates the text to be translated from the format information (RTF, HTML, etc.). Format information can not be encapsulated in the text (as it is done when translating among Romance languages) because large order changes can be produced; therefore, two files are generated from the input: one with the format information and other with the source text for the translation process. In the first file links to the source texts are included. After the analysis phase, *ord* (order of the words in the chunk and of the chunks in the sentence) and *alloc* (position in the analysed text) (see section 3) attributes are added to the output in order to be able to link the translated text from the format file.

2.2. The Spanish analyser

The analyzer of Spanish text is a free available analyzer, *FreeLing* (Carreras et al., 2004), basically a shallow-parser, which has been augmented with a dependency parser. This module links the dependencies among tokens in the chunk, and among chunks in the sentence. The output is an XML structure

where the main elements are the chunks in the sentence, and the nodes (words) in the chunks.

The result includes information about three main elements:

- nodes (lexical form, lemma, POS tag and inflection information)
- chunks (type and dependencies among the words in the chunk)
- sentence (type and dependencies among the chunks in the sentence)

For example in this sentence:

```
porque habré tenido que comer patatas
because I will have had to eat potatoes
```

The output will be (in an interpreted format) the following[1]:

```
subordinate_conjunction: porque[cs]
verb_chain:
   haber[vaif1s]+tener[vmpp0sm]+que[cs]
   +comer[vmn]
noun_chain: patatas[ncfp]
```

2.3. The transfer module

The transfer module is based on the three main elements in the translation process: words or nodes, chunks or phrases and sentences.

First lexical transfer is carried out using a bilingual dictionary compiled into a finite-state transducer using the same format and tool designed for Romance languages (Corbí-Bellot et al., 2005).

Then, structural transfer at the sentence level is applied, and some information is transferred from some chunks to others and some chunks may disappear.

In the previous example the person and number information of the object (third person plural), and the type of subordination (cause) are imported from other chunks to the chunk corresponding to the verb chain.

Finally the structural transfer at the chunk level is carried out. This process is quite simple for noun chains but may be complex for verb chains. A grammar has been developed for this aim (Alegria et al., 2005) and a finite-state transducer is obtained using finite-state tools in the *OpenTrad* project.

The result of this whole process for the previous example is the following:

```
verb_chain:
   jan(main)[partPerf] / behar(per)
   [partPerf] / izan (dum)[partFut] /
   edun(aux)[indPres][subj1s][obj3p]
   +lako[causal]
noun_chain: patata[noun]+[abs][pl]
```

2.4. The generation

The output of the transfer module is passed on to the generation module, where generation is carried out. First, syntactic generation is performed in order to decide the order of the words in the chunks and then morphological generation decides the Basque surface forms basing on the lemmas and morphological information. The main inflection is added to the the last word in the chunk (in Basque: the declension case, the number and other features are assigned to the whole noun phrase at the end of the last word), but in verb chains additional words need morphological generation.

The morphological generation is based on a finite-state transducer based in the previous work of our group (Alegria et al., 1996) but in a standard format compatible with the corresponding tool for translation among Romance languages.

In the example the information between parentheses is used in the syntactic generation phase and the information between brackets in the morphological generation. The final result is the translated sentence, in the case of the example the following:

```
patatak jan behar izango ditudalako
(potatoes eat have-to be-FUT-PAR
   I-have-them-because)
```

2.5. The re-formatter

Finally, the *re-formatter* links the translated text (result of the previous modules), and the format file (saved in the first module), rebuilding a formatted text from the links.

[1] The morphological and syntactical information will be not explained. More details about these informations are shown in (Alegria et al., 2005).

In this process some inconsistencies can be found and the presentation of some documents could be changed.

3. Formats for interaction

A similar DTD specification has been designed to communicate using XML tags the analysis, transfer and generation modules. The main aim is to guarantee the interoperability among the different modules, so that different developers can build or change one or several modules. Although post-editing is not included in the project, the format is able to save enough information for it.

Additionally a format has been specified for the information corresponding to the format of the document to translate.

3.1. The format after the analysis

The output of the analysis is an XML structure where information at sentence, chunk and word level is specified. The dependencies among chunks and among nodes in the chunks are expressed in the structure.

Figure 1 shows the format for the example sentence (*porque habré tenido que comer patatas*) and Figure 2 the corresponding DTD.

It can be observed that a hierarchy system is used in order to explain the dependencies among chunks and among nodes in the chunk. It is a simple but powerful format.

```
<SENTENCE ord="1">
  <CHUNK ord="1" type="conj-subord">
    <NODE ord="1" form="porque" lem="porque" mi="CS" alloc="1"/>
    <CHUNK ord="2" type="grup-verb">
      <NODE ord="4" form="comer" lem="comer" mi="VMN0000" alloc="25">
        <NODE ord="1" form="habré" lem="haber" mi="VAIF1S0" alloc="8"/>
        <NODE ord="2" form="tenido" lem="tener" mi="VMP00SM" alloc="14"/>
        <NODE ord="3" form="que" lem="que" mi="CS" alloc="21"/>
      </NODE>
      <CHUNK ord="3" type="sn" si="obj">
        <NODE ord="1" form="patatas" lem="patata" mi="NCFP000" alloc="31"/>
      </CHUNK>
    </CHUNK>
  </CHUNK>
</SENTENCE>
```

Figure 1.- Output from the analysis module in the example

```
<!ELEMENT        SENTENCE     (CHUNK+)>
<!ATTLIST        SENTENCE
                  ord   CDATA        #REQUIRED
>
<!ELEMENT        CHUNK  (NODE, CHUNK*)>
<!ATTLIST        CHUNK
                  ord         CDATA               #IMPLIED
                  type  (sn|grup-sp|grup-verb|conj-subord|F|...) #REQUIRED
                  si          (subj|obj|...)      #IMPLIED
                  ref         CDATA               #IMPLIED
>
<!ELEMENT        NODE   (NODE*)>
<!ATTLIST        NODE
                  ord         CDATA        #IMPLIED
                  form        CDATA        #IMPLIED
                  lem         CDATA        #REQUIRED
                  pos         CDATA        #IMPLIED
                  mi          CDATA        #REQUIRED
                  ref         CDATA        #IMPLIED
                  alloc       CDATA        #REQUIRED
>
```

Figure 2.- DTD for the output format of the analysis module

```
<SENTENCE ord="1">
  <!-- a CHUNK disappears -->
    <!-- a NODE disappears -->
    <CHUNK type="grup-verb" ref="2">
      <NODE lem="jan" pos="m_verb" mi="PP" ref="4" alloc="25">
        <NODE lem="behar" pos="per" mi="PP" ref="1" alloc="8"/>
        <NODE lem="izan" pos="dum" mi="PF" ref="2" alloc="14"/>
                    <NODE lem="edun" pos="aux" mi="IP_SBJ_OBJ3p+lako_CS" ref="3"
                                                                    alloc="21"/>
      </NODE>
      <CHUNK type="sn" si="obj" ref="3">
        <NODE lem="patata" pos="noun" mi="[abs][pl]" ref="1" alloc="31"/>
      </CHUNK>
    </CHUNK>
</SENTENCE>
```

Figure 3.- Output of the transfer module in the example

```
<SENTENCE ord="1">
  <!-- a CHUNK disappears -->
    <!-- a NODE disappears -->
    <CHUNK ord="2" type="grup-verb" ref="2">
      <NODE ord="1" form="jan" lem="jan" pos="m_verb" mi="PP" ref="4"
                                                                    alloc="25">
        <NODE ord="2" form="behar" lem="behar" pos="per" mi="PP" ref="1"
                                                                    alloc="8"/>
        <NODE ord="3" form="izango" lem="izan" pos="dum" mi="PF" ref="2"
                                                                    alloc="14"/>
        <NODE ord="4" form="ditudalako" lem="edun" pos="aux" mi="IP_SBJ_OBJ3p
                                                    +lako_CS" ref="3" alloc="21"/>
      </NODE>
      <CHUNK ord="1" type="sn" si="obj" ref="3">
        <NODE ord="1" form="patatak" lem="patata" pos="noun" mi="ABS_PL" ref="1"
                                                                    alloc="31"/>
      </CHUNK>
    </CHUNK>
</SENTENCE>
```

Figure 4- Output of the generation module in the example

The attributes *alloc* and *ref* are managed for the recovery of the format in the input text, *mi* is for morphological information and *si* for the syntactical one. The attribute *ord* is used for ordering the elements in the sentence.

3.2. The format after the transfer and generation

The same DTD used for the output of the analysis is also used for the output of the transfer and generation steps, but optional *ref* attribute appears in order to remember the order in the original sentence.

After the transfer (Fig. 3) although slight changes in the structure (some nodes and chunks can be moved) the main changes are produced in the values of the attributes which will be correspond to Basque lemma and morphological or syntactical information instead of the corresponding Spanish information. The *ord* attribute disappears, because a new order will be calculated in the next step. The *form* attribute disappears too, waiting to the morphological generation.

After the generation (Fig. 4) information about order and word-form is added using the

same XML structure.

4. Formats for linguistic data

An adequate documentation of the code and auxiliary files is crucial for the success of open source software. In the case of a MT system, this implies carefully defining a systematic format for each source of linguistic data used by the system. The formats used by the linguistic process have been converted into XML (World Wide Web Consortium, 2004) for interoperability; in particular, for easier parsing, transformation, and maintenance. The XML formats for each type of linguistic data are defined through conveniently-designed XML document-type definitions (DTDs).

On the one hand, the success of the open source machine translation engine heavily depends on the acceptance of these formats by other groups[2]; acceptance may be eased by the use of an interoperable XML-based format which simplifies the transformation of data from and towards it, and also by the availability of tools to manage linguistic data in these formats; the current project is expected to produce transformation and management tools in a later phase. But, on the other hand, acceptance of the formats also depends on the success of the translation engine itself.

There are four sets of linguistic data organized at two levels: lexical or morphological level and structural or syntactical level:

- At lexical level morphological and bilingual dictionaries are used following the proposal for the whole *OpenTrad* project.
- At structural level two grammars are being developed: one for structural transfer and other for syntactical generation.

4.1. Dictionaries

Morphological dictionaries establish the correspondences between surface forms and lexical forms for Basque and contain (a) a definition of the alphabet (used by the tokenizer), (b) a section defining the grammatical symbols used in a particular application to specify lexical forms (symbols representing concepts such as *noun*, *verb*, *plural*, *present*, etc.), (c) a section defining paradigms (describing reusable groups of correspondences between parts of surface forms and parts of lexical forms), and (d) one or more labelled dictionary sections containing lists of surface form—lexical form correspondences for whole lexical units (including contiguous multi-word units). Paradigms may be used directly in the dictionary sections or to build larger paradigms (at the conceptual level, paradigms represent the regularities in the inflective system of the corresponding language).

Bilingual dictionaries have a very similar structure and establish correspondences between source language lexical forms and target language lexical forms, but they seldom use paradigms.

More details about these formats are shown in (Corbí-Bellot et al., 2005).

A Spanish-Basque bilingual dictionary for lexical transfer and a Basque morphological dictionary for morphological generation, both in the proposed format, are included in the MT engine.

4.2. Structural transfer

The format proposed for structural transfer is carried out in two steps: sentence level and chunk level. The process at sentence level is simple and transfers attributes from some chunks to others. Some chunks can disappear. The transference at chunk level is more complex when verb chains have to be managed. A grammar has been written with this aim and good results are obtained (Alegria et al, 2005). The rules of the grammar are declaratives and use regular expressions which are compiled into finite-state transducers.

The format, by the moment, is a text-file composed of regular expressions, and new partners could modify the rules or include a new grammar. The following three types of rules are used[3]:

[2] This is indeed the mechanism by which *de facto* standards appear.

[3] The examples manage verbs which are the most complex chunks for structural transfer.

1. Identification and markup rules. Matches the chunk with its type, adding the corresponding pattern or schema in the target language to the chunk. In this pattern a set of general attributes are included for its substitution in the next step.
   The general format of this rules is the following:

```
[ esVerbChainType @-> ... BORDER
              euVerbChainSchema ]
```

   An schema looks as the next example:

```
(main) Aspm /Per Aspp /Dum Aspd /
       Aux TMood SubjObjDat +RelM
```

2. Attribute replacement rules: These rules replace attributes in the Basque schema with their corresponding values, depending on the values of some attributes in the Spanish verb chain and/or in the Basque schema, which are separated by a BORDER tag. These rules use the left-to-right, longest match conditional replacement operator. For example the following rule:

```
[ Aspp @-> [partFut] || ?* [VMIF|
   VMIC|VAIC] ?* BORDER P1 ?* _ ]
```

   Substitutes the aspect with the participle future in the specified context, which depends on the type of verb in Spanish and Basque.

3. Cleaning rules. Finally two rules remove the unnecessary information, giving the desired output.

This grammar is combined with the structural transfer process (a simpler step) which, at the moment, is solved in a procedural way. In the near future all these steps will be controlled by an unique grammar.

5. Concluding remarks

This paper has shown the current state of development of an open transfer machine translation architecture for Spanish-Basque. This is one of the machine translation engines that will be developed in a large, government-funded open source development project (the other one is a shallow-transfer engine for Romance languages). Furthermore, as a well-documented open source engine, it could be adapted to translating between other languages with few resources to be in the market of MT engines

Some of the components (modules, data formats and compilers) are inherited from previous projects or from the architecture for closer languages, but the format among modules and the grammar for structural transfer are new proposals.

The code, together with pilot Spanish-Basque linguistic data to demonstrate it, will be released at the beginning of 2006 through the project web page (www.opentrad.com).

The principal remaining problem is that the system does not cope with semantic ambiguity. The use of multi-word units in the lexicon relaxes this problem, but we are considering to apply word-sense desambiguation based on the experience in our group (Agirre & Martinez, 2004).

Acknowledgements: Work funded by the Spanish Ministry of Industry, Commerce and Tourism through project *OpenTrad* (FIT-340101-2004-3).

6. References

Agirre E., Martinez D. (2004) The Basque Country University system: English and Basque tasks. Proceedings of the *3rd ACL workshop on the Evaluation of Systems for the Semantic Analysis of Text* (SENSEVAL). Barcelona.

Alegria I., X. Artola, K. Sarasola, M. Urkia (1996). Automatic morphological analysis of Basque. Literary and Linguistic Computing.

Alegria I., A. Diaz de Ilarraza, G. Labaka, M. Lersundi. A. Mayor, K. Sarasola, (2005) A FST grammar for verb chain transfer in a Spanish-Basque MT System. Proc. of the *Finite State Methods in Natural Language Processing workshop*. Helsinki.

Canals-Marote, R., A. Esteve-Guillén, A. Garrido-Alenda, M.I. Guardiola-Savall, A. Iturraspe-Bellver, S. Montserrat-Buendia, S. Ortiz-Rojas, H. Pastor-Pina, P.M. Pérez-Antón, M.L. Forcada (2001). The Spanish-Catalan machine translation system interNOSTRUM,

in *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, 73-76.

Carreras, X., I. Chao, L. Padró and M. Padró (2004). FreeLing: An open source Suite of Language Analyzers, in *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal*.

Corbí-Bellot M., M. L. Forcada, S. Ortiz-Rojas, J. A. Perez-Ortiz, G. Ramirez-Sanchez, F. Sanchez-Martinez, I. Alegria, A. Mayor, K. Sarasola. (2005) An open source Shallow-Transfer Machine Translation Engine for the Romance Languages of Spain. Proceedings of the EAMT2005

Díaz de Ilarraza, A., A. Mayor, K. Sarasola (2000). Reusability of wide-coverage linguistic resources in the construction of a multilingual machine translation system, in *Proceedings of MT 2000 (Univ. of Exeter, UK, 19-22 Nov. 2000)*, .

Garrido, A., A. Iturraspe, S. Montserrat, H. Pastor, M. L. Forcada (1999). A compiler for morphological analysers and generators based on finite-state transducers. Procesamiento del Lenguaje Natural, 25, 93--98.

World Wide Web Consortium (2004). "Extensible Markup Language (XML)", http://www.w3.org/XML/.