Matxin: Moving towards language independence

Aingeru Mayor
IXA Taldea
Euskal Herriko Unibertsitatea
E-20080 Donostia
aingeru@ehu.es

Francis M. Tyers
Dept. Lleng. i Sist. Informàtics,
Universitat d'Alacant
E-03071 Alacant
ftyers@prompsit.com

Abstract

This paper describes some of the issues found when adapting and extending the *Matxin* free-software machine translation system to other language pairs. It sketches out some of the characteristics of *Matxin* and offers some possible solutions to these issues.

1 Introduction

In this paper we present the problems founded when extending the free rule-based machine translation (RBMT) system *Matxin* (Mayor, 2007; Alegria et al., 2007) to work with new language pairs and present the possible solutions.

Matxin is the first publicly available machine translation (MT) system for translating into Basque. The initial aim of the system was to translate into Basque, but its architecture was designed to be independent of both source and target languages.

The *Matxin* 1.0 prototype translates from Spanish to Basque and the development team of the system is currently adapting it to translate from English to Basque pair. The Spanish to Basque prototype is available in two versions. A partially-free version with a full bilingual lexicon is available for testing online[1] and a fully-free version, distributed under the GPL, but with a reduced bilingual lexicon is available for free download.[2]

An independent initiative is looking at implementing a translator using *Matxin* for translation from Breton to English pair. The experience of creating a translator using *Matxin* to translate into a language which is not Basque reveals problems in the current architecture and implementation and gives ideas of how to improve the system with respect to its language independence.

2 *Matxin*: Rule-based machine translation for Basque

The last decade has seen the raise of corpus-based techniques for machine translation, in particular statistical machine translation (SMT) and less research on rule-based techniques. However, translation involving a less-resourced language poses serious difficulties for SMT, as the required large parallel corpora are typically not available. Morphologically-rich languages, such as Basque, have also been shown to be difficult to treat with SMT, as shown in (Koehn and Monz, 2006), where the available SMT systems lag well behind commercial RBMT systems. Having limited digital resources, the rule-based approach is suitable for the development of an MT system for Basque.

Version 1.0 of the system has been evaluated and compared with the state-of-the-art corpus-based *Matrex* MT system (Stroppa et al., 2006; Labaka et al., 2007) translating from Spanish to Basque. The evaluation was performed using the Human-targeted Translation Edit Rate (HTER) metric presented in (Snover et al., 2006; Przybocki et al., 2006), and the comparative results have shown that *Matxin* performs significantly better, with an error rate of 43.60 vs. 57.97 in the parallel corpus on which *Matrex* was trained,

[1] http://www.opentrad.org

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

J.A. Pérez-Ortiz, F. Sánchez-Martínez, F.M. Tyers (eds.)
Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, p. 11–17
Alacant, Spain, November 2009

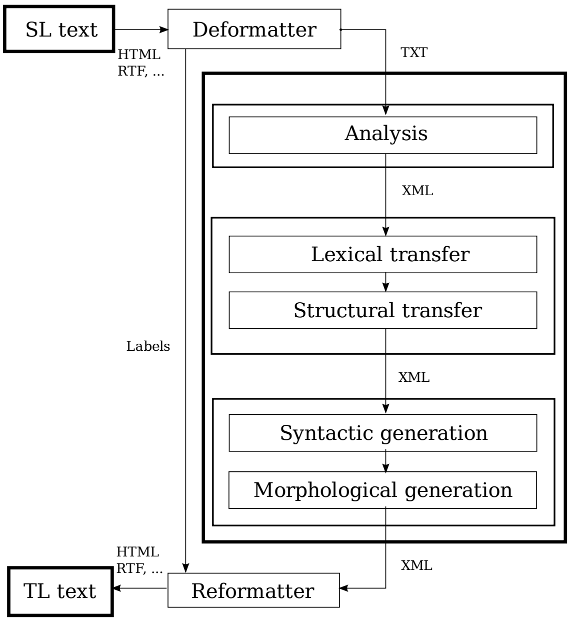

Figure 1: A schematic representation of the architecture.

and 40.41 vs. 71.87 on an out-of-domain corpus. The same system was also evaluated in the framework of *AnHitz* (Arrieta et al., 2008), a 3D virtual expert on science and technology that uses *Matxin* to translate the results of the Cross Lingual Information Retrieval module that are not in Basque. The results of the MT module showed good acceptance to be used for basic understanding. Users of the system found 30% of the translations 'very good', 'good' or 'quite good', and 38.89% of them 'comprehensible'.

These results show that the current version of the system is useful for content assimilation, that is, for understanding a text, but that it is not yet suitable for unrestricted use in text dissemination.

3 System characteristics

The *Matxin* system architecture (see Figure 1) is designed based on the classic transfer-based model (Arnold et al., 1994). The translation process is divided into three phases: analysis, transfer and generation. Other modules have also been added to maintain the format of documents.

Each phase is divided into several modules and translation linguistic tasks have guided the design of the architecture. Linguistic data is separated from algorithms, even the operations required to handle the translation data structure. Monolingual modules are as independent as possible from bilingual modules.

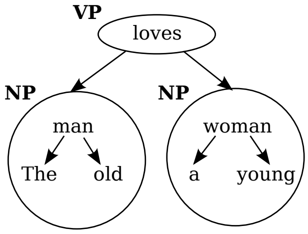

Figure 2: An example of the dependency tree structure: "*The old man loves a young woman*".

Due to the low-resourced status of the Basque language, reuse has played a key part in building the system. Previously developed modules (Spanish analyser, morphological Basque generator, de-formatter and re-formatter, lexical transfer, etc.) and previously generated linguistic resources (dictionaries and corpus) have been reused, and the system has been built so that the modules and the linguistic data created could be reused in other language applications: the Spanish dependency analyser, the dictionary of prepositions, the verbal chunk transfer, etc.

As resource and module heterogeneity is one of the consequences of reuse, to ensure interoperability between reused modules, the linguistic resources and the data flow between modules follow a standard: XML[3].

The dictionaries are coded in a format based on XML according to the *Apertium* project specification (Forcada et al., 2009), and they are converted into binary files using a compiler developed in that project, which creates a representation based on high-performance finite-state transducers.

The data structure which the system processes is based in a hybrid syntactic structure, similar to those described in (Brants et al., 1999; Coheur et al., 2004): the constituents are labelled, and the dependency relations between the words of each of the constituents and between the constituents are expressed (see Figure 2). This data structure uses three types of objects: sentence, chunk and node. A sentence object in the system is equivalent to one sentence and holds a set of chunks. A chunk is broadly equivalent to a constituent

[3] Extensible Markup Language. *http://www.w3.org/XML*

```
<!ELEMENT SENTENCE (CHUNK+)>
<!ATTLIST SENTENCE
 ord    CDATA #IMPLIED <!--Order in the whole text-->
 ref    CDATA #IMPLIED <!--Corresponding SL sentence-->
 alloc  CDATA #IMPLIED <!--Pos. of the 1st character-->
>
<!ELEMENT CHUNK (NODE, CHUNK*)>
<!ATTLIST CHUNK
 ord    CDATA #IMPLIED <!--Order in the sentence-->
 ref    CDATA #IMPLIED <!--Corresponding SL chunk-->
 alloc  CDATA #IMPLIED <!--Pos. of the 1st character-->
 type   CDATA #IMPLIED <!--Chunk type-->
 si     CDATA #IMPLIED <!--Syntactic information-->
 focus  CDATA #IMPLIED <!--Focus-->
 prep   CDATA #IMPLIED <!--Preposition-->
 trans  CDATA #IMPLIED <!--Transitivity-->
 subper CDATA #IMPLIED <!--Subject's person-->
 <!...>
>
<!ELEMENT NODE (NODE*)>
<!ATTLIST NODE
 ord    CDATA #IMPLIED <!--Order in the chunk-->
 ref    CDATA #IMPLIED <!--Corresponding SL node-->
 alloc  CDATA #IMPLIED <!--Position of the 1st character-->
 form   CDATA #IMPLIED <!--Form-->
 lem    CDATA #IMPLIED <!--Lemma-->
 mi     CDATA #IMPLIED <!--Morphological information-->
 pos    CDATA #IMPLIED <!--Part-of-speech-->
 suf    CDATA #IMPLIED <!--Information on the suffix-->
 det    CDATA #IMPLIED <!--Determination-->
 num    CDATA #IMPLIED <!--Number-->
 per    CDATA #IMPLIED <!--Person-->
 loc    CDATA #IMPLIED <!--Location-information-->
<!...>
 >
```

Figure 3: The DTD representing the data structure used for processing sentence translation

and holds a set of nodes, and a node is largely equivalent to a word. The modules of the transfer and generation phases process this data structure and it will be also used for the communication between modules. The format of the data structure is based on XML and represents all the information required to translate the sentences. The DTD (see Figure 3) describes the main elements of the translation process (sentences, chunks and nodes), their attributes[4] and their dependency relations. In the XML data structure one element may contain another element, indicating that it comes below in its dependency structure. An example of the syntactic analysis output by the analyser can be found in Figure 4

```
<?xml version='1.0' encoding='UTF-8' ?>
<corpus>
<SENTENCE ord='1' alloc='0'>
 <CHUNK ord='2' alloc='5' type='grup-verb' si='top'>
  <NODE ord='2' alloc='5' form='es' lem='ser' mi='VSIP3S'>
  </NODE>
  <CHUNK ord='1' alloc='0' type='sn' si='subj'>
   <NODE ord='1' alloc='0' form='Esto' lem='este'
         mi='PD0NS000'>
   </NODE>
  </CHUNK>
  <CHUNK ord='3' alloc='8' type='sn' si='att'>
   <NODE ord='4' alloc='12' form='prueba' lem='prueba'
         mi='NCFS000'>
    <NODE ord='3' alloc='8' form='una' lem='uno'
          mi='DI0FS0'>
    </NODE>
   </NODE>
  </CHUNK>
 </CHUNK>
</SENTENCE>
</corpus>
```

Figure 4: Output of the syntactic analysis for an example sentence *Esto es una prueba* 'This is a test'.

```
abandona abandonar VMIP3S0 abandonar VMM02S0
abandonada abandonar VMP00SF abandonat AQ0FSP
abandonades abandonar VMP00PF abandonat AQ0FPP
abandonant abandonar VMG0000
abandonar abandonar VMN0000
abandonaran abandonar VMIF3P0
abandonarem abandonar VMIF1P0
abandonaren abandonar VMIS3P0
abandonares abandonar VMIS2S0
abandonareu abandonar VMIF2P0
abandonaria abandonar VMIC1S0 abandonar VMIC3S0
abandonarien abandonar VMIC3P0
abandonaries abandonar VMIC2S0
abandonarà abandonar VMIF3S0
abandonaràs abandonar VMIF2S0
abandonaré abandonar VMIF1S0
```

Figure 5: Example of morphology for the verb *abandonar* 'to leave' expressed as a full-form list from the FreeLing morphological dictionary of Catalan.

4 Extending the system to other language pairs

4.1 Morphological analysis

The module for analysis used in *Matxin* is currently FreeLing (Carrera et al., 2008). FreeLing combines morphological analysis, part-of-speech tagging, partial-parsing and dependency analysis into one module. It currently has language data available for eight languages:[5] Asturian, Catalan, Welsh, English, Spanish, Galician, Italian and Portuguese, but only a subset of these have every level of analysis (including chunking and dependency parsing), that is Asturian, Catalan, English and Spanish.

It is fairly straightforward to add a new language pair to FreeLing, each of the modules has separate data files, and in principle no programming is required. However, FreeLing does have problems with morphologically complex languages, for example Basque or Sámi, or languages with productive compounding, for example Icelandic or Norwegian. The morphological analysis module describes the morphology of a

[4] The attributes of the elements express linguistic information and also document format.

[5] http://www.lsi.upc.edu/~nlp/freeling/

language by way of a full-form list in `latin1` encoding, an example of which from the Catalan morphology can be found in figure 5. This works well for languages with low or medium inflection written in the western Latin alphabet, for example English and Spanish, but for Basque, where each word can have many inflected forms, the size of the full-form list becomes unmanageable, and for Sámi, where each word can have many inflected forms, and the alphabet uses characters outside of `latin1` it is even more so. It is worth noting that although FreeLing supports Welsh, it has problems with the characters ŵ and ŷ which are characteristic to Welsh but not found in the `latin1` character encoding.

There are however a number of free toolkits for implementing finite-state morphologies, which have been shown to be effective in the treatment of complex morphologies, for example `foma` (Huldén, 2009) and `hfst` (Lindén et al., 2009). There are also other finite-state toolkits which although not appropriate for complex morphologies have analysers and generators implemented for a wide variety of languages, for example `lttoolbox` (Ortiz-Rojas et al., 2005).

4.2 Dependency analysis

FreeLing allows for both rule-based dependency analysis and data-driven dependency analysis. The data-driven analysis is currently a wrapper for Maltparser (Nivre et al., 2007). There are however, other available systems for doing rule-based dependency parsing, such as VISL Constraint Grammar[6] which have freely-available rule-based parsers available, for example for Faroese (Trosterud, 2009) and North Sámi (Trosterud and Wiechetek, 2007). It could be desirable to make it possible to make the use of Constraint Grammar available as an option in *Matxin*.

4.3 Rules for handling the translation data structure

In the current implementation 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.[7]

The rules written for each of the modules have their formalisms in slightly different formats, although most of them are implemented as tab-delimited files. There are some examples of the rule formalisms in Figure 6.

The rule of the set for interchunk movements in the structural transfer phase expresses that if a chunk has syntactic information (`si`) with the value `subj` (subject), it will pass the information in the attribute `per` (person) to the chunk which is above it (direction `up`) overwriting the contents of the attribute `subjPer`, proving that the chunk is of type `verb-chain`.

The rule for interchunk ordering in the syntactic generation says that chunks (`x2`) that depend on a `verb-chain` chunk (`x1`) and have the focus put in front (`x2.x1`).

Furthermore a few linguistic decisions are expressed in the code of the modules:

- *Matxin* allows basic semantic information to be appended to nouns, for example tags for animacy, or other features. In the current implementation, the tag for noun `[IZE]` is hard-coded in the lexical transfer module. This means that in language pairs which do not use the same tag for nouns, the semantic information is not appended.

- 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, and not expressed by rules.

Along with the abovementioned issues, there is also currently no validation mechanism for the rule files. All of these characteristics make it difficult to modify the rules or to extend the system to new language pairs.

4.4 Proposed rule formalism

To solve these problems in the transfer and generation stages an XML-based format (see Figure 7) has been designed to unify all the different rule

[6] `http://visl.sdu.dk/constraint_grammar.html`

[7] Some specific tasks are done by external modules: search in the bilingual lexicon, translation of prepositions and syntactic functions (Agirre et al., 2009), translation of verbal chunks (Alegria et al., 2005) and morphological generation.

```
  % Rule for interchunk movements (Structural transfer)
  % chunkCondition /originAttrib   chunkCondition    /destinationAttrib  direction  writeMode
    si=’subj’      /per            type=’verb-chain’ /subjPer            up         overwrite

  % Rule for interchunk ordering (Syntactic generation)
  % parentChunkType(x1)   childChunkType(x2)  condition   order
    verb-chain            .*?                 focus=true  x2.x1
```

Figure 6: Examples of rules from the current version of the Spanish–Basque system

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

Figure 7: Rules for handling the data structure: DTD.

formats into a single format. The existing rules (and also those decisions coded in the modules) have been converted by hand (65 rules in total) and may be found in the SVN repository of the *Matxin* project.[8] It is necessary now to develop the interpreter that will apply to the data structure the set of rules corresponding to each module.

Each rule is made up of two parts: the pattern match and the set of actions.

In order to express in the pattern match the configuration to be searched for in the data structure, the definition of the element (or set of elements) of this configuration is indicated. References for previously defined elements can be used when defining each element. Expressions based on the XPath language syntax[9] are used to define elements.

The set of actions contains one or more actions to apply to the elements defined in the pattern match. Actions include the main operations (assignment, deletion, substitution and concatenation) and calls to external functions (searching in the lexicon, using the morphological processor, etc.).

The interpreter, evaluating the XPath expressions defined in the pattern match, collects the references of the elements, and applies the actions specified in the rules to them.

Examples of the rules can be seen in Figure 8.

```
<!--- Copy person-information from subject to
      verb-chunk>
<rule id=’1’>
 <match>
   <def> C1 := //CHUNK[@type=’verb-chain’]</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-chain’]</def>
   <def> C2 := ./CHUNK[@focus=’true’]     </def>
 </match>
 <actions>
   <act> C2/@relord := ’left-jointly’     </act>
 </actions>
</rule>

<!--- 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>
```

Figure 8: Examples of proposed rule formalism

[8] https://matxin.svn.sourceforge.net/svnroot/matxin

[9] http://www.w3.org/TR/xpath/

The first rule defines `C1` as the set of all chunks which are verbal chains (`@type='verb-chain'`), and `C2` as the child chunk of each chunk in the `C1` set, and which contains the syntactic information of the subject value (`@si='subj'`). The actions section indicates that in the subject's person attribute (`@subjper`) for each chunk in the `C1` set we have to copy the person attribute (`@per`) of the corresponding `C2` chunk.

The second rule defines `C1` as the set of all chunks which are verbal chains (`@type='verb-chain'`), and `C2` as the child chunk of each chunk in the `C1` set, and which contains the focus information (`@focus='true'`). The actions section assigns the value `'left-jointly'` to the relative order attribute (`@relord`).

The third rule defines the set of `N` nodes which have not lemma information (`not(@lem)`) and, in the actions, eliminates these nodes.

The fourth rule defines the `N` set of nodes which are nouns (`@pos='[IZE][ARR]'`). We call up the external function `&semDict` for all nodes in the `N` set, with the information of the lemma, and save the result in the semantic information attribute (`@sem`).

This solution will improve the current implementation of the system. It will guarantee that all the linguistic information is coded in declarative rules, and not in the implementation of the system and the unified formalism will make much more easier to add or modify the rules and to create new sets of rules for new language pairs.

5 Discussion

Matxin, the first publicly available MT system for translating into Basque, was designed to be independent of both source and target languages, The experience of creating a translator using *Matxin* to translate into a language which is not Basque has revealed problems in the current architecture and implementation.

In this paper we have described some of the aspects of the *Matxin* system that create problems in adapting the system to new language pairs and proposed some solutions, involving more flexible source language analysis and 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.

6 Acknowledgements

This research was supported in part by the Spanish Ministry of Education and Science (OpenMT, TIN2006-15307-C03-01).

Many thanks to the anonymous reviewers for their valuable and constructive comments.

References

Agirre, E., Atutxa, A., Labaka, G., Lersundi, M., Mayor, A., and Sarasola, K. (2009). Use of rich linguistic information to translate prepositions and grammar cases to Basque. In *EAMT*.

Alegria, I., de Ilarraza, A. D., Labaka, G., Lersundi, M., Mayor, A., and Sarasola, K. (2005). An FST grammar for verb chain transfer in a Spanish-Basque MT system. In *FSMNLP*.

Alegria, I., de Ilarraza, A. D., Labaka, G., Lersundi, M., Mayor, A., and Sarasola, K. (2007). Transfer-based MT from Spanish into Basque: reusability, standardization and open source. In *LNCS 4394. 374-384. Cicling*.

Arnold, D. J., Balkan, L., Meijer, S., Humphreys, R. L., and Sadler, L. (1994). *Machine Translation: an Introductory Guide*. London: Blackwells-NCC.

Arrieta, K., de Ilarraza, A. D., Hernáez, I., Iturraspe, U., Leturia, I., Navas, E., and Sarasola, K. (2008). AnHitz, development and integration of language, speech and visual technologies for Basque. In *Second International Symposium on Universal Communication*, JAPAN.

Brants, T., Skut, W., and Uszkoreit, H. (1999). Syntactic Annotation of a German Newspaper Corpus. In *Proceedings of the ATALA Treebank Workshop*, pages 69–76, Paris, France.

Carrera, J., Castellón, I., Lloberes, M., Padró, L., and Tinkova., N. (2008). Dependency Grammars in FreeLing. *Procesamiento del Lenguaje Natural*, (41):21–28.

Coheur, L., Mamede, N., and Bés, G. G. (2004). From a Surface Analysis to a Dependency Structure. In *Workshop on Recent Advances* *in Dependency Grammar (Coling 2004)*, Ginebra, Suiza.

Forcada, M. L., Bonev, B. I., Rojas, S. O., Ortiz, J. A. P., Sanchez, G. R., Martínez, F. S., Armentano-Oller, C., Montava, M. A., Tyers, F. M., and Rosell, M. G. (2009). Documentation of the Open-Source Shallow-Transfer Machine Translation Platform Apertium. Technical report, Departament de Llenguatges i Sistemes Informatics. Universitat d'Alacant.

Huldén, M. (2009). Foma: a Finite-State Compiler and Library. *EACL 2009*, pages 29–32.

Koehn, P. and Monz, C. (2006). Manual and automatic evaluation of Machine Translation between European languages. In *Proceedings on the Workshop on SMT*, pages 102–121, New York City. ACL.

Labaka, G., Stroppa, N., Way, A., and Sarasola, K. (2007). Comparing Rule-Based and Data-Driven Approaches to Spanish-to-Basque Machine Translation. In *Proceedings of the MT-Summit XI*, Copenhagen.

Lindén, K., Silfverberg, M., and Pirinen, T. (2009). HFST Tools for Morphology - An Efficient Open-Source Package for Construction of Morphological Analyzers. *Proceedings of the Workshop on Systems and Frameworks for Computational Morphology 2009*.

Mayor, A. (2007). *Matxin: Erregeletan oinarritutako itzulpen automatikoko sistema baten eraikuntza estaldura handiko baliabide linguistikoak berrerabiliz*. PhD thesis, University of the Basque Country, Donostia, Euskal Herria.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Ortiz-Rojas, S., Forcada, M. L., and Ramírez-Sánchez, G. (2005). Construcción y minimización eficiente de transductores de letras a partir de diccionarios con paradigmas. *Procesamiento del lenguaje natural*, (35):51–57.

Przybocki, M., Sanders, G., and Le, A. (2006). Edit distance: a metric for Machine Translation evaluation. In *Proceedings of the Fifth LREC*, pages 2038–2043, Genoa, Italy.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the AMTA*.

Stroppa, N., Groves, D., Way, A., and K.Sarasola (2006). Example-Based Machine Translation of the Basque language. In *Proceedings of the 4th LREC*, Boston, USA.

Trosterud, T. (2009). A Constraint Grammar for Faroese. *Workshop on Constraint Grammar, 14th May 2009, Syddansk Universitet, Odense*.

Trosterud, T. and Wiechetek, L. (2007). Disambiguering av homonymi i Nord- og Lulesamisk. *Mémoires de la Société Finno-Ougrienne*, pages 375–395.