Using *foma* for language-based games

Manex Agirrezabal[1], Iñaki Alegria[1], and Mans Hulden[2]

[1] IXA group. University of the Basque Country (UPV/EHU)
manex.aguirrezabal@ehu.es, i.alegria@ehu.es
[2] Ikerbasque (Basque Science Foundation)
mhulden@email.arizona.edu

Abstract. This paper describes two examples of how finite-state technology (FST) commonly used in computational morphology can help implement language-based games. The tool we have used is *foma* an open-source toolkit, similar to previous Xerox/PARC finite-state tools. FST tools have been widely used to describe the morphology of languages and to implement spelling checkers and correctors, especially for highly inflected languages.
This tool can also be used to create language-based games for a large number of languages. Here, we give an account of our experience in developing two games for Basque: (1) Angry Words (*Apalabrados*), for which we generated a list of words, and (2) a tool for making verses in Basque.

1 Introduction

In this paper we describe some examples of the use of finite-state technology (FST) used in computational morphology for implementing language-based games.

Language-based games have always been very popular, and among them, the most popular one is undoubtedly *Scrabble*, a quite simple lexicon-based game, which is played in a lot of countries using different languages. Another popular language-based game is palindromes.

This kind of games are getting more and more popular in educational environments since they are adequate for developing language skills and other competences.

The need for NLP and computational morphology techniques for the development of language-based games is evident. Although some of the lexicon-based games are based on a wordlist, using computational morphology is a more flexible, powerful and general solution. Besides, wordlists are unsuitable for highly inflected languages. Similarly, spelling checking/correction in the 90s was mostly tackled using wordlists, but nowadays morphology-based toolkits are the state-of-the-art techniques.

In the next sections we describe our experience using computational morphology in the development of two games for Basque:

- Angry Words (*Apalabrados*), for which we have generated a list of words. This is a phone multiuser game which follows Scrabble's conventions.

– components for a game for verse makers in Basque. It is based in a previous educational tool named BAD. The key components are: (1) a syllable counter/checker for checking the structure of the verse, and (2) a rhyme finder/checker (Agirrezabal et al., 2012)

The toolkit we use is *foma* an open-source toolkit (Hulden, 2009), similar to previous Xerox/PARC tools based on finite-state morphology (Beesley and Karttunen, 2003). FST tools have been widely used to describe the morphology of languages and are used to implement spelling checkers and correctors, especially for highly inflected languages.

2 *foma*: open-source FST for computational morphology

Finite-State Morphology (Beesley and Karttunen, 2003) has been widely used for word-based applications, especially spelling checking/correction and morphological analysis. A lot of examples for several languages are shown in the above mentioned reference and the corresponding website[3]. The lexicon specification language, *lexc*, is used for modeling the lexicon and constraining the morphotactics. The phonological rules, on the other hand, are aimed at constructing a transducer which models the phonological, morphological and orthographic alternations. These rules are based on regular expressions and are easy to understand. Lexicon and rules are composed and a single transducer is generated for analysis/generation/recognition. Because of the features of this technology the generated tools are very compact and fast.

foma is an FST toolkit licensed under the GNU general public license. Compatibility with the Xerox/PARC toolkit described in (Beesley and Karttunen, 2003) was one of its design goals. The distribution that includes the source code comes with a user manual and a library of examples [4]. The compiler and library are implemented in C and an API is available. The API also contains functionality for finding words that match most closely (but not exactly) a path in an automaton. This makes it straightforward to build spell-checkers from morphological transducers by simply extracting the range of the transduction and matching words approximately. Evaluation scores show that *foma* seems to perform comparably to the Xerox/PARC toolkit.

There have been developed different tools using *foma*. In figure 1 you can see an example script using this notation. This script performs the syllabification. Each language have special features, so it may not be completely correct for all worldwide languages.

The original description for Basque (Alegria et al., 1996) used in the examples was adapted to the *foma* toolkit (Alegria et al., 2010). Two versions were obtained. The first one was morphology-oriented and contained the whole lexicon containing and a full morphological description of all the morphemes (including category, case, tense, person...). The second version was spelling-oriented and

[3] http://www.stanford.edu/ laurik/fsmbook

[4] https://code.google.com/p/foma/

```
define V [a|e|i|o|u];
define Gli [w|y];
define Liq [r|l];
define Nas [m|n];
define Obs [p|t|k|b|d|g|f|v|s|z];

define Onset (Obs) (Nas) (Liq) (Gli); # Each element is optional.
define Coda  Onset.r;                 # Is mirror image of onset.

define Syllable Onset V Coda;
regex Syllable @> ... "." || _ Syllable;
```

Fig. 1. Some rules for "Angry Words"

only included the accepted words. In both descriptions the same phonological rules were used.

3 First game: *Apalabrados* for Basque

This game works based on a limited list of words for each language so we were asked to build up a wordlist for the Basque version of the game.

Although Basque is an agglutinative language with a rich morphology, the list had to be obviously limited. Therefore, in order to constrain its size, we used our previous morphological description for Basque together with some rules.

For the first version of the game only most frequent nouns, verbs and adjectives were included and the following restrictions were applied (some of the rules for these restrictions are shown in figure 2):

- for nouns: only 22 basic declension cases for each root
- for adjectives: only the root and declension for graduation (comparative, superlative)
- for non-finite verbs: 24 forms for each, including 3 participles and some declension cases for nominalization
- for auxiliary and synthetic verbs: only their bare form and the relative, completive and causative suffix
- prefixes were excluded

For these rules *foma* was used writing rules for intersection of the whole description of the language and morphological patterns.

This first version has been included by the company[5] and it has already been made available. An example appears in figure 3.

[5] www.apalabrados.com/

```
# No prefixes
define NoPrefix ~$[ Prefix1 | Prefix2 | ... ] ;
# Auxiliary and sintetic verbs:
#   only the basic form and relative, completive and causative derived
define ConstVerbs
    $[ AuxSintVerbs ( ?* [ RelatSuff | ComplSuff | DerivSuff ])];
...
# Composition
define Verbs ConstVerbs .o. NoPrefix .o. AllWordsMorph ;
define ConsVerbList Verbs.l ;
```

Fig. 2. Some rules for “Angry Words”

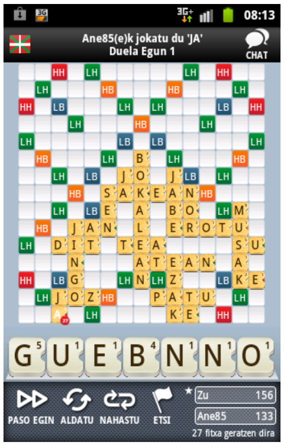

Fig. 3. Example of interface of playing Angry Words in Basque

We are planning to prepare a second version with a wider vocabulary and improved constraints. Some proposals from the players can be read on *Sustatu*[6] webpage (in Basque).

In the future we want to explore the use of the FST technology in order to build a tool to help players find the best solutions taking into account the present patterns and tokens.

4 Second game: Verse making

Bertsolari tradition is a form of improvised verse composition and singing, where participants are asked to produce impromptu compositions about themes which are given to them following one of many alternative verse formats.

[6] http://sustatu.com/euskaljakintza/1344882156

Verses in the *bertsolari* tradition must consist of a specified number of lines, each with a fixed number of syllables. Also, strict rhyme patterns must be followed. The structural requirements are considered to be the most difficult element, however, well-trained *bertsolaris* can usually produce verses that fulfill the structural constraints in a very limited time.

Based on FST technology, we developed some components for verse making in Basque. A game is currently under development based on the components used for an educational tool named BAD (Agirrezabal et al., 2012).

In the game, just like in the Basque bertsolari tradition, verse-makers must compose stanzas [7] in turns in order to assemble a whole verse. Each player will get a score according to the quality of the stanza they have composed. On the first version, the game will evaluate mainly the metrics of the stanzas (number of syllables and rhymes). For a second version, we intend to evaluate the semantic and pragmatic quality of the rhymes/verses too.

Potential target users for the game are *bertso-eskolak*, schools aimed at training young people in the verse-making art. These schools have grown in popularity and can currently be found throughout the Basque Country.

The components of this tool which are based on FST technology are (1) a syllable counter/checker for checking verse structure, and (2) a rhyme finder/checker.

The syllable counter/checker performs syllabification based on the approach described in (Hulden, 2006), with some modifications to capture Basque phonology. It also has a module that deals with variants produced by optional apocope and syncope phenomena.

In the rhyme checker some FST rules are applied. The first rule identifies the segments that do not participate in the rhyme and marks them off with { and } symbols (e.g. landa–ganga → `<{l}anda>-<{g}anga>`). Another rule removes everything that is between { and }, leaving us only with the segments relevant for the rhyming pattern (e.g. `<anda>-<anga>`). Subsequent to this rule, a phonological grouping reduction is applied producing, for example (`<aNMBDGRa>-<aNMBDGRa>` where NM and BDGR are metacharacters.

After this reduction, we use the *eq(X,L,R)-* operator in *foma*, which from a transducer X, filters out those words in the output where material between the specified delimiter symbols L and R are unequal. In our case, we use the `<` and `>` symbols as delimiters, yielding a final transducer that does not accept non-rhyming words.

A component for searching words that rhyme with a given word is also integrated in the tool. It uses a finite-state component like the one developed with foma. Similarly to the techniques previously described, it relies on extracting the segments relevant to the rhyme, after which phonological rules are applied to yield phonetically related forms. For example, introducing the pattern “era“, the system returns four phonetically similar forms “era“, “eda“, “ega“, and “eba“. Then, these responses are fed to a transducer that returns a list of words with

[7] A stanza is a kind of division of a poem. It is composed of two or more lines and usually its syllabic and rhyming structure is repeated throughout the verse.

the same endings. To check the endings with the dictionary we use the described finite-state morphological description of Basque.

Acknowledgments

This research has been partially funded by the Spanish Ministry of Education and Science (OpenMT-2, TIN2009-14675-C03) and the Basque Government (Research Groups, IT344-10).

Thanks to the reviewers for their interesting comments and proposals for improvements. We must acknowledge Maite Goñi for her proposal of localizing the Angry Words game to Basque.

Bibliography

Agirrezabal, M., Alegria, I., Arrieta, B., and Hulden, M. (2012). BAD: An assistant tool for making verses in basque. page 13.

Alegria, I., Artola, X., Sarasola, K., and Urkia, M. (1996). Automatic morphological analysis of Basque. *Literary & Linguistic Computing*, 11(4):193–203.

Alegria, I., Etxeberria, I., Hulden, M., and Maritxalar, M. (2010). Porting basque morphological grammars to foma, an open-source tool. *Finite-State Methods and Natural Language Processing*, pages 105–113.

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

Hulden, M. (2006). Finite-state syllabification. *Finite-State Methods and Natural Language Processing*, pages 86–96.

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