From Portuguese to Mirandese: Fast Porting of a Letter-to-Sound Module Using FSTs

Isabel Trancoso[1], Céu Viana[2], Manuela Barros[2], Diamantino Caseiro[1], and Sérgio Paulo[1]

[1] L[2]F - Spoken Language Systems Lab
INESC-ID/IST
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
{Isabel.Trancoso,dcaseiro,spaulo}@l2f.inesc-id.pt
http://www.l2f-inesc-id.pt/
[2] CLUL
Av. Prof. Gama Pinto 2, Lisbon, Portugal
{mcv,manuela.barros}@clul.ul.pt
http://www.clul.ul.pt/

Abstract. This paper describes our efforts in porting our letter-to-sound module from European Portuguese to Mirandese, the second official language in Portugal. We describe the rule formalism and the composition of the various transducers involved in the letter-to-sound conversion. We propose a set of extra SAMPA symbols to be used in the phonetic transcription of Mirandese, and we briefly cover the set of rules and results obtained for the two languages. Although at a very preliminary stage, we also describe our efforts at building a waveform generation module also based on finite state transducers. The use of finite state transducers allowed a very flexible and modular framework for deriving and testing new rule sets. Our experience led us to believe that letter-to-sound modules could be helpful tools for researchers involved in the establishment of orthographic conventions for lesser spoken languages.

1 Introduction

Mirandese is the smallest language spoken in the Iberian peninsula. It is spoken by a population that does not exceed 12,000 and covers only a region of 500 square kilometres, on the northeastern border of the country. It is a romance language, related to Asturian-Leonese, and for several centuries it was preserved only as an oral transmission language. Its recognition as official language is fairly recent (1999) and so are the efforts to create an orthographic convention [1] in order to establish unifying criteria for writing in this language[1].

The motivation for deriving letter-to-sound rules for Mirandese (*Mirandés*), was to build a tool that may help native speakers to learn how to read and write, as well as students interested in that language.

[1] http://mirandes.no.sapo.pt

N.J. Mamede et al. (Eds.): PROPOR 2003, LNAI 2721, pp. 49–56, 2003.

As a starting point, we used the rules that we had derived for European Portuguese (EP). The first letter-to-sound module that we developed for EP was in the context of a rule-based system (DIXI). In fact, none of the data-driven tools that we had developed since then (either based on neural networks [2] or *CARTs* - Classification and Regression Trees [3]) were suited for Mirandese, given the small amount of training material.

Our most recent efforts in terms of letter-to-sound conversion were based on Finite State Transducers (*FSTs*) [4], motivated by their flexibility in integrating multiple sources of information and other interesting properties such as inversion. The knowledge-based approach using *FSTs* is flexible enough to allow easy porting to similar languages or other varieties of Portuguese.

This paper describes our efforts in porting our *FST*-based letter-to-sound module from European Portuguese (EP) to Mirandese. We start by the description of the rule formalism (Sect. 2) and of the composition of the various transducers involved in the letter-to-sound conversion (Sect. 3). We proceed with the proposal of a set of SAMPA symbols to be used in the phonetic transcription of Mirandese (Sect. 4). The next two sections present the main results for EP and Mirandese. Finally, we describe our preliminary efforts at building other modules of a concatenative-based synthesizer using *FSTs* (Sec 7) and present some concluding remarks.

2 Rule Formalism

In our first rule-based system for EP (DIXI [5]), the rules were written in the usual form $\phi \rightarrow \psi / \lambda \_ \_ \rho$ where ϕ, ψ, λ and ρ can be regular expressions that refer to one or multiple levels. The meaning of the rules was the following: when ϕ was found in the context with λ on the left and ρ on the right, ψ would be applied, replacing it or filling a different level of ψ. Most of the grapheme-to-phone rules were written such that ϕ, λ and ρ only referred to the grapheme level (with stress marks already placed on it) and ψ only to the phone level. There were no intermediate stages of representation and no rule created or destroyed the necessary context for the application of another rule. In order to prevent some common errors, a small set of 6 rules was nevertheless added which referred to grapheme-phone correspondences on either context λ or ρ.

Although some similarities may be found between DIXI's and a Two-Level Phonology approach ([6], [7]), DIXI's rules were not two-level rules: contexts were not fully specified as strings of two-level correspondences and within the set of rules for each grapheme, a specific order of application was required. Default rules needed to be the last and in some cases in which the contexts of different rules partially overlapped, the most specific rule needed to be applied first.

Our first step in the design of the *FST*-based rule system was to convert DIXI's rules to a set of *FSTs*. In order to preserve the semantic of these rules we opted to use rewriting rules, but in the following way:

First, the grapheme sequence $g_1, g_2, ..., g_n$, is transduced into $g_1, \_, g_2, \_, ..., \_, g_n$, where $\_$ is an *empty* symbol, used as a placeholder for

phones. Each rule will replace _ with the phone corresponding to the previous grapheme, keeping it. The context of the rules can now freely refer to the graphemes. The few DIXI rules whose context referred to phones can also be straightforwardly implemented. In this way, we avoid rule dependencies that would be necessary if we had just replaced graphemes by phones: the first rule would only have graphemes in its context, while the last ones have mainly phones. The very last rule removes all graphemes, leaving a sequence of phones. The input and output language of the rule transducers is thus a subset of $(grapheme\ phone)^*$. The set of graphemes and the set of phones do not overlap.

The rules are specified using a rule specification language, whose syntax resembles the BNF (Backus Naur Form) notation, allowing the definition of non-terminal symbols (e.g. `$Vowel`). Regular expressions are also allowed in the definition of non-terminals. Transductions can be specified by using the *simple transduction* operator $a \rightarrow b$, where a and b are terminal symbols. This work motivated us to extend the language with two commands.

The first command is:

$$\texttt{OB\_RULE}\ n,\ \phi \rightarrow \psi / \lambda \_\_\_ \rho$$

where n is the rule name and $\phi, \psi, \lambda, \rho$ are regular expressions. `OB_RULE` specifies a context dependent *obligatory rule*, and is compiled using Mohri and Sproat's algorithm [8].

The second one is:

$$\texttt{CD\_TRANS}\ n,\ \tau \Rightarrow \lambda \_\_\_ \rho$$

where τ is a transducer (an expression that might include the $\rightarrow$ operator). `CD_TRANS` (Context-Dependent Transduction) is a generalization where the replacing expression depends on what was matched. It is compiled using a variation of Mohri and Sproat's algorithm, that uses $\pi_1(\tau)$ instead of ϕ, and τ instead of the cross product $\phi \times \psi$. Its main advantage is that it can succinctly represent a set of rules that apply to the same context. We use it mainly in the stress-marking phase.

3 Transducer Composition

The rules of the letter-to-sound module are organized in various phases, each represented by transducers that can be composed to build the full module. Figure 1 shows how the various phases are composed.

Each phase has the following function:

- `introduce-phones` is the simple rule that inserts the _ *empty phone* placeholder after each grapheme. (`$Letter (NULL → EMPTY)`) ⇒ ___).
- the `exception-lexicon` contains the pronunciation of frequent words not covered by the rules.
- the `stress` phase consists of rules that mark the stressed vowel of the word.
- `prefix-lexicon` consists of pronunciation rules for compound words, namely with roots of Greek or Latin origin such as "tele" or "aero".

```
introduce-phones  o
exception-lexicon o
     stress       o
 prefix-lexicon   o
     gr2ph        o
     sandhi       o
remove-graphemes
```

Fig. 1. Phases of the knowledge based system

- `gr2ph` is the bulk of the system, and consists of a set of rules that convert the graphemes (differentiating between diacritics) to phones.
- `sandhi` implements word co-articulation rules across word boundaries. (This rule set was not tested here, given the fact that the test set consists of isolated words.)
- `remove-graphemes` removes the graphemes in order to produce a sequence of phones. (`$Letter → NULL / ___`).

The following example (in EP) illustrates the specification of 2 `gr2ph` rules for deriving the pronunciation of grapheme *g*: either as /Z/ (e.g. *agenda, gisela*) when followed either by *e* or *i*, or as /g/ otherwise (SAMPA symbols used).

```
OB_RULE 0200, g EMPTY -> g _Z \
    / NULL ___ ($AllE | $AllI)

OB_RULE 0201, g EMPTY -> g _g \
    / NULL ___ NULL
```

The compilation of the rules may result in a very large number of *FSTs* that may be composed in order to build a single grapheme-to-phone transducer. Alternatively, to avoid the excessive size of this single transducer, one can selectively compose the *FSTs* in order to obtain a smaller set that can be later composed with the grapheme *FST* in runtime to obtain the phone *FST*.

4 The SAMPA Phonetic Alphabet for Both Languages

The SAMPA phonetic alphabet for EP[2] was defined in the framework of the SAM_A European project and includes 38 phonetic symbols. Table 1 lists the additional symbols that had to be defined for Mirandese, together with some examples. They cover two nasal vowels, 3 non-strident fricatives corresponding to b, d, g in intervocalic position or after r, and 2 retroflex fricatives.

[2] `http://www.l2f.inesc-id.pt/~imt/sampa.html`

Table 1. Additional SAMPA symbols for Mirandese

SAMPA	Orthography	Transcription
@~	centelha	s@~t"ejL6
E~	benga	b"E~g6
B	chuba	tS"uB6
D	roda	R"OD6
G	pega	p"EG6
s_	sol	s_"Ol~
z_	rosa	R"Oz_6

5 Transducer Approach for European Portuguese

The transducer approach for EP involved a large number of rules: 27 for the `stress` transducer, 92 for the `prefix-lexicon` transducer, and 340 for the `gr2ph` transducer. The most problematic one was the latter. We started by composing each of the other phases into a single *FST*. `gr2ph` was first converted to a *FST* for each grapheme. Some graphemes, such as *e*, lead to large transducers, while others lead to very small ones. Due to the way we specified the rules, the order of composition of these *FSTs* was irrelevant. Thus we had much flexibility in grouping them and managed to obtain 8 transducers with an average size of 410k. Finally, `introduce-phones` and `remove-graphemes` were composed with other *FSTs* and we obtained the final set of 10 *FSTs*.

In runtime, we can either compose the grapheme *FST* in sequence with each *FST*, removing dead-end paths at each step, or we can perform a lazy simultaneous composition of all *FSTs*. This last method is slightly faster than the DIXI system.

In order to assess the performance of the *FST*-based approach, we used a pronunciation lexicon built on the PF ("Português Fundamental") corpus. The lexicon contains around 26,000 forms. 25% of the corpus was randomly selected for evaluation. The remaining portion of the corpus was used for training or debugging. As a reference, we ran the same evaluation set through the DIXI system, obtaining an error rate of 3.25% at a word level and 0.50% at a segmental level.

The first test of the *FST*-based approach was done without the *exception lexicon*. The *FST* achieved almost the error rate of the DIXI system it is emulating, both at a word level (3.56%) and at a segmental level (0.54%). When we integrate the exception lexicon used in DIXI, the performance is exactly the same as for DIXI. We plan to replace some rules that apply to just a few words with lexicon entries, thus hopefully achieving a better balance between the size of the lexicon and the number of rules.

6 Transducer Approach for Mirandese

The porting of the *FST*-based approach from EP to Mirandese involved changing the `stress` and `gr2ph` transducers. The stress rules showed only small differences

compared to the ones for EP (e.g. stress of the words ending in *ç*, *n*, and *ie*). The `gr2ph` transducer was significantly smaller than the one developed for EP (around 100 rules), reflecting the much closer grapheme-phone relationship.

The hardest step in the porting effort involved the definition of a development corpus for Mirandese. Whereas for EP the choice of the reference pronunciation (the one spoken in the Lisbon area and most often observed in the media), was fairly easy, for Mirandese it was a very hard task, given the differences between the pronunciations observed in the different villages of the region. This called for a thorough review of the lexicon, and checking with native speakers. For development, we used a small lexicon of about 300 words extracted from the examples in [1]. For testing, we used a manually transcribed lexicon of around 1,100 words, built from a corpus of oral interviews conducted by CLUL in the framework of the ALEPG project (*Atlas Linguístico-Etnográfico de Portugal e da Galiza*). As a starting point, we selected the interviews collected in the village of Duas Igrejas, which was also the object of the pioneering studies of Mirandese by José Leite de Vasconcelos [9].

Our first tests were done without an exceptions lexicon. In our very small development set, we obtained 11 errors (3.68% error rate at a word level), all of which are exceptions (foreign words, function words, etc.). For the test set, a similar error rate was obtained (3.09%). Roughly half of the errors will have to be treated as exceptions, and half correspond to stress errors. For more details concerning differences between the two rule sets, and a discussion of the types of error, see [10].

7 FST-Based Concatenative Synthesis

This section describes on-going work toward the development of other modules of a text-to-speech (TTS) system using *FSTs*. In particular, it covers the waveform generation module, which is based on the concatenation of diphones.

A diphone is a recorded speech segment that starts at the steady phase of a first phone (generally close to the mid part of the phone) and ends at the steady phase of the second one. By concatenating diphones, one can capture all the events that occur in the phone transitions, which are otherwise difficult to model.

Our *FST*-based system is in fact based on the concatenation of triphones which builds on this widely used diphone concatenation principle. A triphone is a phone that occurs in a particular left and right context. For example, the triphone *a-b-c* is the version of *b* that has *a* on the left and *c* on the right. In order to synthesize *a-b-c*, we concatenate the diphones *a*—*b* and *b*—*c* and then remove the portions corresponding to phones *a* and *c*.

Our first step in the development of this type of system for EP was the construction of a diphone database. A common approach is to generate a set of nonsense words (logathomes), containing a center diphone as well as surrounding carrier phones. After generating the list of prompts, they were recorded in a sound proof room, with a head mounted microphone to keep the recording con-

ditions reasonably constant among sessions. We also tried to avoid variations on the speaker's rhythm and intonation, in order to reduce concatenation problems.

The following step was the phonetic alignment of the recorded prompts, which was made manually. Rather than marking the phone boundaries, we need to select phone mid parts. For each triphone *a-b-c*, we tried to minimize discontinuities on both diphones *a—b* and *b—c*, by performing a local search for the best concatenation point in the mid parts of the two samples of *b*. We used the Euclidean distance between the Line Spectral Frequencies (LSF), because of their relationship to formant frequencies and their bandwidths. By avoiding discontinuities on the formants, we solve some of the concatenation problems, but not all of them. Since at the chosen points for concatenation, the signal energy may differ, the last step is to scale the speech signals at the diphone boundaries. The scale factor is the ratio between the energy of the last pitch period of the first diphone and the energy at the first pitch period of the second diphone. This scale factor will approach one as we approach the phone boundary, to avoid changing the energy of other phones. We were not very concerned with the discontinuities of the signal fundamental frequency, because, during the recording procedure, the speaker kept it pretty constant.

Using the triphone database, speech synthesis can be performed by first converting graphemes into phones, then phones into triphones, and finally concatenating the sound waves corresponding to the triphones. This process can be represented as the transducer composition cascade $W \circ G2P \circ Tr \circ DB$, where W is the sentence, $G2P$ is the grapheme-to-phone transducer, Tr is the phone-to-triphone transducer and finally DB is a transducer that maps triphones into sequences of samples.

The phone-to-triphone transducer Tr is constructed as the composition of two bigram transducers $Tr = B_{di} \circ B_{ph}$. The bigram transducers map their input symbols into pairs of symbols, for example, given a sequence a, b they produce $(\#, a), (a, b), (b, \#)$.

The bigram transducer can be built by creating a state for each possible input symbol and creating, for each symbol pair (a, b), an edge linking state a with state b with input b and output (a, b).

This prototype system, which, for the time being is completed devoided of prosody modules, was only build for EP. However, the system can be used with the Mirandese letter-to-sound transducer composed with a phone mapping transducer in order to produce an approximation of the acoustic realization of an utterance in Mirandese as spoken by an EP speaker. We expect to have funding in the near future to record a native Mirandese speaker and process the corresponding database.

8 Concluding Remarks

This paper described an *FST*-based approach to letter-to-sound conversion that was first developed for European Portuguese and later ported to the other official language in Portugal - Mirandese. The hardest part of this task turned out to

be the establishment of a reference pronunciation lexicon that could be used as the development corpus, given the observed differences in pronunciation between the inhabitants of the small villages in that region.

The use of finite state transducers allows a very flexible and modular framework for deriving new rule sets, and testing the consistency of the orthographic conventions. Based on this experience, we think that letter-to-sound systems could be useful tools for researchers involved in the establishment of orthographic conventions for lesser spoken languages. Moreover, such tools could be helpful in the design of such conventions for other partner languages in the CPLP community.

Acknowledgments. We gratefully acknowledge the help of António Alves, Matilde Miguel, and Domingos Raposo.

References

1. M. Barros-Ferreira and D. Raposo, editors. *Convenção Ortográfica da Língua Mirandesa*. Câmara Municipal de Miranda do Douro – Centro de Linguística da Universidade de Lisboa, 1999.
2. I. Trancoso, M. Viana, F. Silva, G. Marques, and L. Oliveira. Rule-based vs. neural network based approaches to letter-to-phone conversion for portuguese common and proper names. In *Proc. ICSLP '94*, Yokohama, Japan, September 1994.
3. L. Oliveira, M.C. Viana, A.I. Mata, and I. Trancoso. Progress report of project dixi+: A portuguese text-to-speech synthesizer for alternative and augmentative communication. Technical report, FCT, January 2001.
4. D. Caseiro, I. Trancoso, L. Oliveira, and C. Viana. Grapheme-to-phone using finite state transducers. In *Proc. 2002 IEEE Workshop on Speech Synthesis*, Santa Monica, CA, USA, September 2002.
5. L. Oliveira, M. Viana, and I. Trancoso. A rule-based text-to-speech system for portuguese. In *Proc. ICASSP '1992*, San Francisco, USA, March 1992.
6. K. Koskenniemi. *Two-Level morphology: A general Computational Model for Word-Form Recognition and Production.* PhD thesis, University of Helsinki, 1983.
7. E.L. Antworth. Pc-kimmo: A two-level processor for morphological analysis. Technical report, Occasional Publications in Academic Computing No 16. Dallas, TX: Summer Institute of Linguistics, 1990.
8. M. Mohri and R. Sproat. An efficient compiler for weighted rewrite rules. In *34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, USA, 1996.
9. J. Vasconcellos. *Estudos de Philologia Mirandesa*. Imprensa Nacional, Lisboa, 1900.
10. D. Caseiro, I. Trancoso, C. Viana, and M. Barros. A comparative description of gtop modules for portuguese and mirandese using finite state transducers. In *Proc. ICPhS' 2003*, Barcelona, Spain, August 2003.