APPLYING TRANSDUCERS TO SPOKEN LANGUAGE PROCESSING FOR PORTUGUESE

Isabel Trancoso, Diamantino Caseiro

L^2F Spoken Language Systems Lab.
INESC-ID/IST
Rua Alves Redol 9, 1000-029 Lisbon, Portugal
{Isabel.Trancoso, dcaseiro}@l2f.inesc-id.pt

ABSTRACT

This paper has two different goals. The primary aim is to illustrate the advantages of weighted finite state transducers for spoken language processing, namely in terms of their capacity to efficiently integrate different types of knowledge sources. We have chosen three areas to emphasize several aspects of the application of transducers: large vocabulary continuous speech recognition, automatic alignment and grapheme-to-phone conversion. The secondary goal is to simultaneously present the state of the art in these areas for European Portuguese.

1. INTRODUCTION

The goal of this paper is two-fold. The primary aim is to illustrate the advantages of weighted finite state transducers (*WFSTs*) for spoken language processing, namely in terms of their capacity to efficiently integrate different types of knowledge sources.

By attempting to illustrate these advantages for several areas of spoken language processing (recognition, alignment, and synthesis), we hope to be able to simultaneously present some of the state of the art in these areas for our language in our lab. This secondary goal is also important in the context of the current presentation aimed at the spoken language processing community in Spain.

There are many possible applications of *WFSTs* for spoken language processing. The three that we have chosen as topics for this paper attempt to emphasize different aspects of transducers.

- In terms of *LVCSR*, our focus will be on how to do the composition of knowledge sources dynamically, with limited memory requirements, and being able to preserve the original sources.
- In terms of alignment, our emphasis will be on the flexible way in which additional knowledge sources such as alternative pronunciation rules may be integrated.
- In terms of speech synthesis, we shall restrict our presentation to grapheme-to-sound conversion, a task where we hope to illustrate the benefits of combining both knowledge-based and data-driven approaches via transducers.

2. LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

The integration of knowledge sources in large vocabulary continuous speech recognition using weighted finite state transducers is spreading in the speech recognition community. Among the main advantages of the approach relative to traditional systems are: the elegant and uniform formalism that allows very flexible ways of integrating multiple knowledge sources, and the superior search performance obtained when the search network is optimized using automata determinization and minimization.

The main disadvantages are related to the search space optimization, both in terms of memory requirements and dynamic adaptation. In fact, the *WFST* determinization algorithm, based on subset construction, and normally used during optimization, requires large amounts of memory relative to the size of the resulting *WFST*. Moreover, although the optimized search network is not much larger than the language model (typically only 2 to 2.5 times larger), it can still be very large and requires large amounts of memory in runtime.

The fact that the optimization of the search network is performed offline means that the original knowledge sources are not available at runtime. Thus, it may be troublesome to preserve the optimality of the network, when dynamically adjusting the knowledge sources. For example, when adapting the language model probabilities or when adding new words or pronunciations to the vocabulary.

The most common solution proposed for the memory requirement problems consist of reducing the size of the system to the resources available during development or run time, and rely on an additional pass to re-score with a larger language model. This approach is very effective, but it only avoids the problem, it does not solve it. Along the same lines, but relying on a single pass, is the approach of Dolfing and Hetherington [1], where the full language model (G_f) is factorized into a small model (G_s) and a difference model (G_{f-s}) such that $G_f = G_s \circ G_{f-s}$; the small language model is incorporated with the other knowledge sources, the lexicon L and the acoustic model H, in a network ($N = H \circ L \circ G_s$) that is optimized offline. The full language model information is integrated in runtime by composing the network with the difference language model ($N \circ G_{f-s}$) “on-the-fly”.

This section describes how we address these problems in our system. In particular, we shall focus on our lexicon and language model “on-the-fly” composition algorithm which, together with a memory efficient representation for *WFSTs* and other language model optimizations [2], allowed us to extend the *WFST* approach to larger systems, such as the ones used for broadcast news recognition.

2.1. Lexicon and Language Model Composition

The determinization of the composition of the lexicon L with the language model G is probably the most resource intensive subtask when optimizing the search network. The reason lies on the large size of the language model, and on the fact that, when applying the subset-construction determinization algorithm, every state of the resulting *WFST* ($det(L \circ G)$) corresponds to a *set* of states of the non-deterministic $L \circ G$ transducer.

In [3] we presented a memory-efficient specialized algorithm for the composition of the lexicon with the language model. Our algorithm is based on Mohri's theorem [4] that states that the composition of sequential transducers is also sequential. This important result means that if we determinize both the lexicon and the language model, then we only need to compose them to obtain the deterministic composition. In practice, we cannot just apply the usual composition algorithm [5], because of ϵ labels on the output tape of the lexicon, which will generate so many non-coaccessible[1] paths in the result, as to make the method unpractical.

Our algorithm is just a specialized composition algorithm, and works as follows: in a preprocessing step, the set of reachable non-ϵ output labels is associated with each ϵ-output edge of the lexicon. That set is used during composition to avoid the generation of non-coaccessible paths by only following ϵ-output edges on the lexicon that will lead to a non-ϵ label compatible with labels in the language model state.

This basic algorithm was also extended to allow output-label and weight pushing. In [6] we showed how to extend the algorithm to approximate "on-the-fly" minimization.

When used with a caching scheme, the overhead of performing both the $LG = L \circ G$ specialized composition and the $H \circ LG$ composition in runtime, is as low as 6% of the search effort[2].

2.2. Application to Broadcast News Recognition

All the recognition experiments described in this section were based on the broadcast news corpus collected in the scope of the ALERT European project[7].

The acoustic models are based on the combination of the output of various neural networks [8]. We extracted 3 different sets of features from the speech signal: 12 *PLP*coefficients + log energy + deltas; 12 log-rasta coefficients + log energy + deltas; and 28 modulation spectrogram features.

We used 3 separate multilayer perceptrons (*MLP*), one for each set of features. The input of each *MLP* was a window of 7 vectors centered on the vector being analyzed. The *MLPs* had a 3-layer architecture with 1000-4000 units in the hidden layer, and the output consisted of 40 softmax units corresponding to 38 context independent phones plus silence and inspiration noise. The output of the 3 *MLPs* was combined using the average of the logarithm of the probability estimated for each phone.

The acoustic model topology consisted of a sequence of states with no self-loop to enforce the minimal duration of the model, and one final state with a self-loop. The acoustic models were encoded in a single acoustic model *WFST*.

We used an European Portuguese lexicon with 57k words and 4-gram backoff language models, trained from more than 384 million words from newspaper texts and interpolated with models obtained from broadcast news transcriptions.

Align	MLP	Data	Decoder	F0	All	xRT
5	1000	22h	Stack	18.3	33.6	30.0
“	“	“	WFST beam 5.5	18.7	32.0	6.4
6	“	46h	“	18.7	31.6	4.1
7	“	“	“	18.8	31.6	4.8
“	2000	“	“min det L	18.0	30.7	4.3
“	4000	“	“	16.9	29.1	3.7

Table 1. Results with BN recognition in successive alignment passes.

Table 1 shows the results obtained with a development test set of 6h, in successive alignment passes [9] (first column) as we changed several parameters in the system: number of units in the hidden layer (second column), number of hours of training data (third column), and type of decoder (fourth column). The word error rates are shown both for F0 and all conditions (fifth and sixth columns) together with the corresponding speed (rightmost column).

The most impressive result is the change in real-time performance observed when we changed from our previous stack decoder to the *WFST* based implementation, obtained without degradation in terms of WER.

3. ALIGNMENT

An automatic alignment module can be used for different purposes. In our past research, we have use such modules as a step in the bootstrap process of training better models for ASR and also as a step to segment better units for concatenative speech synthesis.

This section starts with the description of how we modified our *WFST*-based decoder to be used as an aligner. Two types of alignment were considered: word-based and phone-based alignment. The first is specially important for the alignment of spoken books [10], the first application which we will briefly describe. The second is specially important for aligning spontaneous speech in dialogs. This application is also interesting because of two issues: the large amount of cross-talk and pronunciation variation [11].

The cross-talk problem is specially severe in overlapping turns, but even in non-overlapping conditions, the amount of cross-talk observed was enough to yield very bad alignment results. We thus tried to decrease the observed cross-talk between the two channels, by using source separation techniques, as a pre-processing stage, before doing the alignment.

The pronunciation variation problem is much more important in the spontaneous speech data than in the read speech data of the spoken book alignment application. This motivated us to investigate the possibility of adding a new knowledge source to our search space - alternative pronunciation rules.

3.1. Decoder modifications for alignment purposes

Our aligner is based on *WFSTs* in the sense that its search space is defined by a distribution-to-word (or distribution-to-phone) transducer that is built outside the decoder. For the alignment task, that search space is usually build as $H \circ L \circ W$, where H is the phone

[1] A non-coaccessible path is a "dead-end" paths that does not reach a final state.

[2] The time spent evaluating the distributions (neural network or Gaussian mixtures) is not included in this percentage.

topology, L is the lexicon and W is the sequence of words that constitutes the orthographic transcription of the utterance. As no restrictions are placed on the construction of the search space, it can easily integrate other sources of knowledge, and can be optimized and replaced by an optimal equivalent one.

In order to cope with possible de-synchronizations between the input and output labels of the *WFST*, the decoder was extended to deal with special input labels that are internally treated as epsilon labels (similar to skip arcs in Hidden Markov Models), but are used to mark time transitions or boundaries. Whenever such end-of-segment labels are crossed, the time is recorded in the current hypothesis. The user may choose to place those labels at the end of each phone *WFST* or at the end of each word *WFST*, depending on choosing either phone-level alignment or word-level alignment, respectively.

3.2. Alternative pronunciation rules

In doing the alignment, instead of building a lexicon with multiple pronunciations per word, we may choose to use phonological rules together with a lexicon of canonical forms, in order to account for alternative pronunciations.

These rules are specified using a finite-state grammar whose syntax is similar to the Backus-Naur-form (BNF) augmented with regular expressions. We added the operator $\rightarrow$, simple transduction, to the usual set of operators, such that ($a \rightarrow b$) means that the terminal symbol a is transformed into the terminal symbol b. The language allows the definition of non-terminal symbols (e.g. *$vowel*). All rules are optional, and are compiled into *WFSTs*. We do not apply the rules one by one on a cascade of compositions, but, because they are optional rules, we rather build their union in order to avoid the exaggerated growth of the resulting transducer, which can be exponential with the length of the composition cascade. The rule transducer R is thus build as $R = \bigcup_i \Sigma^*(R_i\Sigma^*)^*$ where R_i is the transducer corresponding to a particular rule specification expression. The rules are applied as $H \circ R^{-1} \circ R^{-1} \circ R^{-1} \circ L \circ W$, where R^{-1} is the inverse of the rule transducer. The rule transducer R is used three times in order to reduce the dependency on the order of the rules. An example of a sandhi rule specification is:

```
$V = $Vowel | $NasalVow | $Glide | $NasalGli;
DEF_RULE S_z, ($V (S -> z) WORD_BREAK $V)
```

3.3. Application to spoken book alignment

The main goal of spoken book alignment is to improve the access to digitally stored spoken books, used primarily by the visually impaired community, by providing tools for easily detecting and indexing units (words, sentences, topics). Simultaneously, we also aimed to broaden the usage of multimedia spoken books (for instance in didactic applications, etc.), by providing multimedia interfaces for access and retrieval.

A small pilot corpus (*O Senhor Ventura*, by Miguel Torga) was chosen as a test bed for spoken book alignment. The high-quality DAT recordings were manually edited to remove reading errors and extraneous noises, amounting to a total of 2h15m (around 138k words, corresponding to 5k different forms). Although very intelligible, as expected from a professional speaker, the speaking rate was relatively high - 174 words per minute.

A major advantage of our approach is that it allowed us to align the full audio version of the book in a single step. This is specially important if we take into account that the memory limitations of our previous alignment tool imposed a maximum of 3-minute audio segments. We thus avoid the tedious task of manually breaking-up the audio into smaller segments with their associated text. The word segmentation of the book took 197.5 seconds in a 600MHz Pentium III computer (0.024 xRT), and required 200MB of RAM.

Although this particular application required only word-level alignment, from the point of view of research, indexed spoken books provide an invaluable resource for data-driven prosodic modeling and unit selection in the context of text-to-speech synthesis, thus motivating us to apply phone-level alignment to this corpus as well.

3.4. Application to the alignment of spontaneous speech in dialogs

Coral is a map task dialog corpus, involving spontaneous conversations between pairs of speakers about map directions. In the 16 different pairs of maps, the names of the landmarks were chosen to allow the study of some connected speech phenomena, such as for instance, sequences of plosives formed across word boundaries (e.g. *clube de tiro*).

The recordings involved 32 speakers (students from the Lisbon area), and took place in a small sound proof room at INESC. The two speakers were separated by a distance of about one meter with a small screen wall in between them, whose goal was to avoid direct visual contact between the participants, but did not provide acoustic isolation. The speakers wore close-talking microphones and the recordings were made in stereo directly to DAT and later down-sampled to 16 kHz per channel.

All dialogs were orthographically transcribed following the same transliteration conventions using SGML format of other map task corpora[3]. Because of the large amount of cross-talk observed, we adopted an adaptive noise canceling scheme [12], as a pre-processing stage.

The experimental results described in this section were obtained with the acoustic models trained for a broadcast news recognition task which were briefly described above. These models cannot yet adequately model for instance laughs and certain filled pauses which are so frequent in the Coral dialog corpus.

Since we only have one pilot dialog manually annotated with time stamps for word boundaries, and not for phone boundaries, our results refer only to word level tests. We started by measuring the average absolute error between the reference time stamps and the automatic ones for each word start, without using either channel separation or alternative pronunciation rules. The lexicon, which we shall denote by Lex0, includes only canonical forms. Multiple pronunciations are exclusively used for heterophonic homographs (amounting to 21).

For the left channel, the average error was 0.380s. For the right channel, the average error was 2.346s (first line of table 2). The larger errors obtained with the latter speaker can perhaps be due to much smaller turns, many of them grunts largely overlapping with the other speaker's turns. Without using channel separation, we observe that the end of the turn is not properly detected, which causes words from one of the speakers to be frequently aligned during the other speaker's turn. The problem is aggravated when overlap occurs. When channel separation is used, the average error decreases as shown in the second line of the same table. The

[3] http://www.hcrc.ed.ac.uk/dialogue/maptask.html

alignment obtained with the separated signals is fairly good. An analysis of the largest errors shows they may be due to the fact that we did not try to align laughs, which causes severe misalignments in the neighboring words. The performance in overlapping turns is on the same level as the one in non overlapping turns.

Next we investigated the relevance of providing alternative pronunciations for function words and forms of the verb *estar*, which were so frequently marked with micro-annotations in our corpus. The values obtained with this new lexicon (Lex1, including multiple pronunciations for 40 forms) are shown in the third and fourth lines of table 2, without and with channel separation respectively. Given these results, further tests with alternative pronunciation rules were done only with Lex1 and channel separation.

The main phonological aspects that alternative pronunciation rules are intended to cover are: (1) intra-word vowel devoicing; (2) voicing assimilation; and (3) vowel and consonant deletion and coalescence. Both (2) and (3) may occur within and across word boundaries. Some common contractions are also accounted for, with both partial or full syllable truncation and vowel coalescence. Vowel reduction, including quality change, devoicing and deletion, is specially important for European Portuguese, being one of the features that distinguishes it from Brazilian Portuguese and that makes it more difficult to learn for a foreign speaker. As a result of vowel deletion, rather complex consonant clusters can be formed across word boundaries. Even simple cases, such as the coalescence of the two plosives in *grade de ferro*, raise interesting problems of whether they may be adequately modeled by a single acoustic model for /d/.

The results obtained with 56 rules are shown in the last line of the table. Phone alignment error would be a much more adequate measure, but unfortunately, we do not yet have reference labels. We observed that the misalignments due to the absence of models of laughs, although affecting only the neighboring words, can be as large as 5s, and almost destroy any potential improvements brought by the use of the rules. For the left channel, for instance, only 1.7% of the word boundary errors are above 1s and most of these errors are due to such segments. The next step will clearly be marking them and creating adequate acoustic models. In order to do this for the whole corpus, automatic alignment followed by posterior manual correction is crucial. Whereas the alignment obtained with the original signals without channel separation is too bad to serve as a starting point, the one obtained with channel separation seems good enough.

Av. error [s]	Ch. sep.	Left-ch.	Right-ch.
Lex0 / no rules		0.380	2.346
Lex0 / no rules	√	0.097	0.151
Lex1 / no rules		0.343	2.334
Lex1 / no rules	√	0.086	0.146
Lex1 / rules	√	0.077	0.143

Table 2. Average word alignment error.

4. GRAPHEME-TO-PHONE CONVERSION

The last part of this paper is devoted to the description of a grapheme-to-phone conversion module based on *WFSTs* for European Portuguese. We investigated both the use of knowledge-based and data-driven approaches.

The objective of a grapheme-to-phone module implemented as *WFSTs* is justified by their flexibility in the efficient and elegant integration of multiple sources of information, such as the information provided by other "text-analysis" modules. The flexibility of *WFSTs* also allows the easy integration of knowledge-based with data-driven methods.

Our first approach to grapheme-to-phone (GtoP) conversion for European Portuguese was a rule-based system (DIXI), with about 200 rules[13]. All the code was programed in C, directly in the case of the stress assignment rules, and using the SCYLA ("Speech Compiler for Your Language") [14] rule compiler, developed by CSELT, for the remaining rules. The multi-level structure of this compiler allowed each procedure to simultaneously access the data resulting from all the previous procedures, so the rules could simultaneously refer to several levels (such as the grapheme level, phone level, sandhi level, etc.).

In this section, we first show how we compiled the rules of the DIXI system to *WFSTs*. We then present data-driven approaches to the problem, and finally we combine the knowledge-based with the data-driven approaches [15].

In order to assess the performance of the different methods, we used a pronunciation lexicon built on the PF ("Português Fundamental") corpus. The lexicon contains around 26000 forms. 25% of the corpus was randomly selected for evaluation. The remaining portion of the corpus was used for training or debugging.

The size of the training material for the data-driven approaches was increased with a subset of the BD-Público [16] text corpus. This corpus includes a collection of texts from the on-line edition of the Público newspaper. We used all the words occurring in the first 1,000,000 paragraphs of this corpus, and obtained their transcription by rule using DIXI. The 205k words not in PF were added to the training set.

4.1. Knowledge-Based System

Our first goal was to convert DIXI's rules to a set of *WFSTs*. SCYLA rules are of 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 is that when ϕ is found in the context with λ on the left and ρ on the right, ψ will be applied, replacing it or filling a different level of ψ.

In order to preserve the semantic of DIXI's rules we opted to use rewriting rules, but, to avoid unnecessary rule dependencies due to the replacement of graphemes by phones, we used them 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. 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 the language described in 3.2. This work motivated us to extend the language with commands such as:

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[17].

The rules of the grapheme-to-phone system are organized in various phases, each represented by transducers that can be composed to build the full system. Figure 4.1 shows how the various phases are composed. Each phase has the following function:

- the `stress` phase consists of 27 rules that mark the stressed vowel of the word.
- `introduce-phones` is the simple rule that inserts the _ *empty phone* placeholder after each grapheme. ($Letter (NULL → EMPTY)) ⇒ ___).
- `prefix-lexicon` consists of pronunciation rules for compound words, namely with roots of Greek or Latin origin such as "tele" or "aero". It includes 92 rules.
- `gr2ph` is the bulk of the system, and consists of 340 rules, that convert the 45 graphemes (including graphically stressed versions of vowels) to phones.
- `sandhi` implements word co-articulation rules across word boundaries. (This rule set was not tested here, given the fact that the test set consists of isolated words.)
- `remove-graphemes` removes the graphemes in order to produce a sequence of phones. ($Letter → NULL / ___).

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

Fig. 1. Phases of the knowledge based system.

The following example 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 results in a very large number of *WFSTs* (almost 500) that need to be composed in order to build a single grapheme-to-phone transducer. We did not build a single *WFST* but selectively composed the *WFSTs* and obtained a small set of 10 *WFSTs* that are composed with the grapheme *WFST* in runtime to obtain the phone *WFST*.

The most problematic phase was `gr2ph`. We started by composing each of the other phases in a single *WFST*. `gr2ph` was first converted to a *WFST* 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 *WFSTs* 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 *WFSTs* and we obtained the final set of 10 *WFSTs*.

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

We evaluated the *WFST*-based rule approach, and compared its performance with the one of our previous rule-based DIXI system. As can be seen in table 3, the *WFST* achieved almost the error rate of the DIXI system it is emulating, both at a word level and at a segmental level. The two rightmost columns show the error rates obtained without taking stress mark errors into account. The difference between the performance of the current and previous approaches is due to the *exception lexicon* included in DIXI that we did not yet implement. We plan to integrate this lexicon and balance its size with the rule system, in order to simplify it by replacing rules that apply to just a few words with lexicon entries.

System	*% Error*		*% Error w/o stress*	
	word	*segm.*	*word*	*segm.*
WFST	3.56	0.54	3.13	0.47
DIXI	3.25	0.50	2.99	0.45

Table 3. Comparison of the current and previous rule-based approaches.

4.2. Data-Driven Approach

The first step in preparing the training corpus for the data-driven techniques consisted of aligning each grapheme with the corresponding phone. We performed the alignment by minimizing the string-edit distance between corresponding grapheme and phone strings, obtaining a sequence of pairs (grapheme, phone), where the grapheme or the phone can both be ϵ. Our first data-driven approach consisted of modeling that sequence using an n-gram model, as proposed by [18].

This model is based on the probability of a grapheme matching a particular phone given the history up to the previous $n-1$ pairs $(P((g_i,p_i)|(g_{i-n-1},p_{i-n-1})...(g_{i-1},p_{i-1})))$.

The language model is first converted to a finite-state acceptor (*WFSA*) over pairs of symbols, and then to a finite-state transducer t, by transforming each pair of symbols into an input and an output label. t is ambiguous because epsilons are used to model backoff transitions during the conversion from n-gram to *WFSA*, and hence, even is there is an explicit n-gram in the model, the *WFSA* will still allow alternative paths that use the backoff.

Due to this ambiguity, in order to use the *WFST* to convert a grapheme sequence *WFST* g to phones, we need to compute $bestpath(\pi_2(g \circ t))$.

We trained various n-gram backoff language models using history lengths $n-1$ ranging from 2 to 7. Table 4 shows the size of the various models, and table 5 shows the error rate on the test set (second and third columns).

4.3. Combining Data-Driven and Knowledge-Based Approaches

One of the greatest advantages of the *WFST* representation is the flexible way in which different methods may be combined. In this

n	*n-grams*	*states*	*edges*	*bytes*
8	1,392,426	820,778	1,983,113	42M
7	981,565	592,184	1,459,738	30M
6	657,107	361,944	980,123	20M
5	401,855	159,425	549,398	11M
4	173,307	37,869	208,668	4M
3	42,451	3,618	46,018	0.8M

Table 4. Pair n-gram *WFSTs*.

n	*% Error*		*% Error w/o stress*	
	word	*graph.*	*word*	*graph.*
8	9.04	1.37	6.11	0.90
7	9.02	1.37	6.12	0.90
6	9.16	1.37	6.13	0.90
5	9.86	1.46	6.38	0.93
4	15.34	2.25	9.23	1.32
3	31.62	4.62	18.42	2.67

Table 5. Performance of the n-gram approach.

section we show some examples of the combination of data-driven with knowledge-based methods.

In [18], as an example of the integration of knowledge-based with data-driven methods, some improvements were obtained by composing the n-gram *WFST* with a *WFST* that restricts the primary stress to exactly one per word. This type of restriction had also been implemented in a neural network based approach that we developed [19] as a post-processing filter.

We opted for a different approach: as we have the stress marking *WFST* stress, we decided to perform the grapheme-phone alignment of the training data not with the original words, but with the output of the stress *WFST*. The alignments thus obtained were used to build n-gram *WFSTs*, as described in section 4.2. To convert a sequence of graphemes g to phones, we now use $bestpath(\pi_2(g \circ \texttt{stress} \circ t))$. Table 6 shows the results obtained with this variation with several n-gram models. We observe a reduction of the word error rate to less than half. The result is even more impressive when we remember that around 90% of the training set was converted by rule with a system that has around 3% errors. The size of the n-gram *WFSTs* was similar.

n	*% Error*		*% Error w/o stress*	
	word	*graph.*	*word*	*graph.*
8	4.01	0.61	3.65	0.54
7	3.94	0.59	3.58	0.53
6	4.02	0.61	3.66	0.55
5	4.04	0.61	3.68	0.55
4	4.48	0.67	4.13	0.60
3	6.40	0.96	6.15	0.91

Table 6. Performance of the combined approach.

5. CONCLUDING REMARKS

This paper attempted to illustrate the potential of *WFSTs* for spoken language processing. This potential is leading us to apply transducers to yet more areas such as unit selection and text normalization in concatenative speech synthesis, and also speech-to-speech translation.

At the same time as pursuing these ambitious goals, we have not forgotten many open issues in the three topics we have investigated so far.

In terms of *LVCSR*, we are currently working on extending our system to include larger lexica and language models, and also on how to incorporate more sophisticated knowledge sources.

In terms of alignment, the whole investigation of alternative pronunciation rules is still dependent on the existence of manually labeled spontaneous speech data, so that we can test the effectiveness of different types of rule. This manual labeling process is now currently being done using as a starting point the automated labels achieved so far. In spite of this open issue, the robustness of the *WFST*-based aligner was fully demonstrated first by its application to spoken books and later by its application to channel-separated dialogs.

In terms of GtoP, the number of open issues is also too large. We plan to improve our rule-based approach by obtaining a better balance between number of rules and lexicon size, as explained earlier. We also plan to convert our *CART*-based approach to the *WFST* framework. This will give us much flexibility in combining the various methods, for example, a *WFST* resulting from the conversion of the tree of a particular grapheme could replace the respective grapheme rules in the *WFST* rule-based system.

The inversion property of transducers opens the possibility of using GtoP techniques in tasks such as reconstructing out of vocabulary words [20] in large vocabulary speech recognition systems. This is an area which we also plan to explore in the near future. One last goal in this long list is the development of GtoP modules for other varieties of Portuguese.

6. ACKNOWLEDGMENTS

This paper includes contributions from many members of the L2F group whom we would like to acknowledge: Hugo Meinedo, João Paulo Neto, Céu Viana, Luís Oliveira, António Serralheiro, Isabel Mascarenhas and also Fernando Silva from the Neural Networks group of INESC ID. Last but not the least, we would also like to thank Antonio Rubio for his invitation to give this presentation, thus allowing us to show our most recent progress in spoken language processing for Portuguese to our Spanish colleagues and friends. INESC-ID Lisboa had support from the POSI program of the “Quadro Comunitario de Apoio III”.

7. REFERENCES

[1] H. Dolfing and I. Hetherington, “Incremental Language Models for Speech Recognition Using Finite-State Transducers,” in *Proc. ASRU ’2001 Workshop*, Madonna di Campiglio, Trento, Italy, December 2001.

[2] D. Caseiro and I. Trancoso, “Using Dynamic WFST Composition for Recognizing Broadcast News,” in *Proc. ICSLP ’2002*, Denver, USA, Sept. 2002.

[3] D. Caseiro and I. Trancoso, “On Integrating the Lexicon with the Language Model,” in *Proc. Eurospeech ’2001*, Aalborg, Denmark, September 2001.

[4] M. Mohri, “Finite-State Transducers in Language and Speech Processing,” *Computational Linguistics*, vol. 2, no. 23, 1997.

[5] M. Mohri, F. Pereira, and M. Riley, “Weighted Automata in Text and Speech Processing,” in *ECAI 96 Workshop*. Budapest, Hungary, August 1996.

[6] D. Caseiro and I. Trancoso, “Transducer Composition for “On-the-Fly” Lexicon and Language Model Integration,” in *Proc. ASRU ’2001 Workshop*, Madonna di Campiglio, Trento, Italy, December 2001.

[7] H. Meinedo, N. Souto, and J. Neto, “Speech Recognition of Broadcast News for the European Portuguese Language,” in *Proc. ASRU ’2001 Workshop*, Madonna di Campiglio, Trento, Italy, December 2001.

[8] H. Meinedo and J. Neto, “Combination of Acoustic Models in Continuous Speech Recognition Hybrid Systems,” in *Proc. ICSLP ’2000*, Beijing, China, October 2000.

[9] H. Meinedo and J. Neto, “Automatic Speech Annotation and Transcription in a Broadcast News Task,” in *submitted to ISCA Workshop on Multilingual Spokne Document Retrieval*, Macau, China, April 2003.

[10] D. Caseiro, H. Meinedo, A. Serralheiro, I. Trancoso, and J. Neto, “Using WFSTs for Aligning Spoken Books,” in *Proc. HLT 2002 - Human Language Technology Conference*, San Diego, California, March 2002.

[11] D. Caseiro, F. Silva, I. Trancoso, and C. Viana, “Automatic Alignment of Map Task Dialogs Using WFSTs,” in *Proc. ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation*, Aspen, CO, USA, Sept. 2002.

[12] B. Widrow, J. Glover, J. McCool, C. Williams, R. Hearn, J. Zeidler, Dong E., and R. Goodlin, “Adaptive Noice Canceling: Principles and Applications,” *Proceedings of the IEEE*, vol. 63, pp. 1692–1716, Dec. 1975.

[13] 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.

[14] S. Lazzaretto and L. Nebbia, “Scyla: Speech Compiler for your Language,” in *Proc. of the European Conf. on Speech Technology*, Edimburgh, UK, September 1987, vol. 2.

[15] D. Caseiro, I. Trancoso, L. Oliveira, and C. Viana, “Graphem-toPhone Using Finite State Transducers,” in *Proc. 2002 IEEE Workshop on Speech Synthesis*, Santa Monica, CA, USA, September 2002.

[16] J. Neto, C. Martins, H. Meinedo, and L. Almeida, “The Design of a Large Vocabulary Speech Corpus for Portuguese,” in *Proc. Eurospeech ’97*, Rhodes, Greece, Sept. 1997.

[17] 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.

[18] R. Sproat, “Corpus-Based Methods and Hand-Build Methods,” in *Proc. ICSLP ’2000*, Beijing, China, October 2000.

[19] 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, Sept. 1994.

[20] B. Decadt, J. Duchateau, W. Daelemans, and P. Wambacq, “Transcription of Out-of-Vocabulary Words in Large Vocabulary Speech Recognition Based on Phoneme-to-Grapheme Conversion,” in *Proc. ICASSP’2002*, Orlando, Florida, May 2002.