Finite-state transducer inference for a speech-input Portuguese-to-English machine translation system

David Picó, Jorge González, Francisco Casacuberta [*] *Diamantino Caseiro, Isabel Trancoso* [†]

Dept. de Sist. Informàtics i Computació
Universitat Politècnica de València
València, Spain

L^2F Spoken Language Systems Lab.
INESC-ID/IST
Lisbon, Portugal

Abstract

Statistical techniques and grammatical inference have been used for dealing with automatic speech recognition with success, and can also be used for speech-to-speech machine translation. In this paper, new advances on a method for finite-state transducer inference are presented. This method has been tested experimentally in a speech-input translation task using a recognizer that allows a flexible use of models by means of efficient algorithms for on-the-fly transducer composition. These are the first reported results of a speech-to-speech translation task involving European Portuguese input that we know of.

1. Introduction

Automatic speech recognition (ASR) has traditionally benefited of major advances thanks to the use of statistical techniques that have been also extended for the development of speech-to-speech machine translation (SSMT) systems [6, 15, 7, 10]. Under this framework, an SSMT system is built from sets of examples that must be sufficiently large and representative. These examples usually consist of parallel text and speech data of the source language.

Two interesting techniques from this field are *statistical alignments* [4, 15] and *finite-state transducers* (FST) [12, 2, 5]. Statistical aligments model mappings between sequences of words from a source language to a target language and have been developed for machine translation during the last two decades [4]. Finite-state transducers have found a wide range of applications, due to their simplicity and the possibility of combining them easily with conventional models in ASR (such as hidden Markov models).

Another important feature of FSTs is that they allow for the implementation of the two basic architectures: sequential (or *decoupled*: first speech decoding and next text translation) and integrated (speech decoding and translation are performed in the same process).

In section 2 we introduce the notation of finite-state transducers used throughout the paper. Section 3 describes a general framework for transducer inference and two translation algorithms. An integrated SSMT architecture is presented in section 4, and experimental results on a Portuguese-to-English task are presented in section 5.

2. Finite-state models

A *weighted finite-state automaton* (WFSA) is a tuple $\mathcal{A} = (\Gamma, Q, i, f, P)$, where Γ is an alphabet of symbols, Q is a finite set of states, functions $i : Q \rightarrow \mathbb{R}$ and $f : Q \rightarrow \mathbb{R}$ give a weight to the possibility of each state to be an initial or final state, respectively, and parcial function $P : Q \times \{\Gamma \cup \lambda\} \times Q \rightarrow \mathbb{R}$ defines a set of transitions between pairs of states in such a way that each transition is assigned a weight and it is labeled with a symbol from Γ or with the empty string, λ.

A *weighted finite-state transducer* (WFST) [14] is defined similarly to a weighted finite-state automaton, with the difference that transitions between states are labeled with *pairs* of symbols that belong to the Cartesian product of two different (*input* and *output*) alphabets, $(\Sigma \cup \{\lambda\}) \times (\Delta \cup \{\lambda\})$.

When weights are probabilities, and under certain conditions, a WFSA can define a distribution of probabilities on the free monoid. In this case it is called a *stochastic* finite-state automaton.

Weighted automata and transducers are able to associate a weight to each accepted string or pair of strings. In the particular case of stochastic models, transition weights are probabilities and the final weight associated to a string or pair of strings is also a probability. Given some strings $\bar{x}$ and $\bar{y}$, an automaton $\mathcal{A}$ and a transducer $\mathcal{T}$, we denote these probabilities as $P(\bar{x}|\mathcal{A})$ and $P(\bar{x}, \bar{y}|\mathcal{T})$.

The possibility of using WFSAs and WFSTs to model weighted languages and translations makes them useful as a formalism for representing different kinds of information involved in different parts of a speech-to-speech translation system, such as acoustic information, lexical

[*] This work has been partially supported by the Spanish project TIC 2003-08681-C02-02.

[†] This work has been partially supported by Portuguese project POSI/PLP/47175/2002.

information, etc. A fully finite-state-based translation system can be assembled from a collection of weighted finite-state models that are made to work together by means of some combining operation, such as composition.

A transducer $\mathcal{T}$ can be understood as implementing a relation $T(\mathcal{T})$ between the input and the output alphabets. The *composition* of two WFSTs, $\mathcal{T} \circ \mathcal{T}'$, is a transducer that implements the composition of their relations.

In the case of *stochastic* WFSTs, we can define the following weight function:

$$W(\bar{x}, \bar{z} | \mathcal{T} \circ \mathcal{T}') = \sum_{\forall \bar{y} \in \Delta^\star} P(\bar{x}, \bar{y} | \mathcal{T}) P(\bar{y}, \bar{z} | \mathcal{T}').$$

The function $W(\bar{x}, \bar{z} | \mathcal{T} \circ \mathcal{T}')$ is not a distribution of probabilities because the composed transducer may not be stochastic. However, in many instances a stochastic transducer can be obtained from $\mathcal{T} \circ \mathcal{T}'$ by normalization, see [9].

3. GIATI: a framework for transducer inference

In general, modelling languages is an easier task than modelling translations. While many useful algorithms for learning finite-state automata (or equivalent models) have been proposed, the literature about the inference of finite-state transducers is much more reduced. Such an algorithm is the GIATI method [5] summarized below .

Given a parallel corpus consisting off a finite sample A of string pairs $(\bar{x}, \bar{y}) \in \Sigma^\star \times \Delta^\star$:

1. Each training pair $(\bar{x}, \bar{y})$ from A is transformed into a string $\bar{z}$ from an *extended alphabet* Γ yielding a sample S of strings, $S \subset \Gamma^\star$.

2. A (stochastic) finite-state automaton $\mathcal{A}$ is inferred from S.

3. Edge symbols in $\mathcal{A}$ are transformed back into pairs of strings of source/target symbols (from $\Sigma^\star \times \Delta^\star$), thus transforming it into a transducer $\mathcal{T}$.

The first transformation is modeled by some labeling function $\mathcal{L} : \Sigma^\star \times \Delta^\star \to \Gamma^\star$, while the last transformation is defined by an "inverse labeling function" $\Lambda(\cdot)$, such that $\Lambda(\mathcal{L}(A)) = A$.

The purpose of building a corpus of strings out of a bicorpus of string pairs in step 2 of the previous method is to condense somehow the meaningful information that we can extract about the relations laying between the words in the input and output sentences. Discovering these relations is a problem that has been thoroughly studied in statistical machine translation and has well-established techniques for dealing with it. The concept of *statistical alignment* [3] formalizes this problem. An alignment is a correspondence between words from an input text to words from an output text. Whether this is a one-to-one, a one-to-many or a many-to-many correspondence depends on the particular definition that we are using. Constraining the definition of alignment simplifies the learning but subtracts expressive power to the model. The available algorithms try to find a compromise between complexity and expressiveness.

3.1. Two translation algorithms

GIATI as defined above is a general framework for designing transducer inference algorithms. Let us describe two different translation algorithms following this framework. In order to explain them more clearly, we will use a tiny example of a English-to-Portuguese alignment: We will consider that the English phrase *the configuration program* is aligned with the Portuguese phrase *o programa de configuração* with the alignment $\{1 \to 1, 2 \to 4, 3 \to 2\}$, i.e., the first source word is aligned with the first target word; the second source word is aligned with the forth target word and the third source word is aligned with the second target word.

3.1.1. Algorithm #1: using a language of segment pairs

1. Transform string pairs into strings: For each pair $(\bar{x}, \bar{y})$ in the sample, the composed string is a sequence of $|\bar{x}|$ pairs, $(u_i, \bar{v}_i)$, where $u_i = x_i$ and $\bar{v}_1 \bar{v}_2 \ldots \bar{v}_{|\bar{x}|} = \bar{y}$. Each of these pairs is considered to be *a single symbol*. We refer the reader to [5] for a complete description of how these pairs are extracted from the alignments and other minor details. Applying this algorithm to the alignment of our example would produce the following corpus containing one single string:

   S = {(the, o) (configuration, λ) (program, programa de configuração)}

2. Infer a finite-state automaton: a smoothed n-gram model can be inferred from the corpus of strings obtained in the previous step. Such a model can be expressed in terms of a WFSA [11].

3. Undo the transformation: a transducer can be obtained directly by considering each of the compound symbols not as a single token, but as the pair of strings that constitute the label of a transition in a transducer.

3.1.2. Algorithm #2: using a corpus of bilingual phrases

1. Transform string pairs into strings: this transformation also obtains a set of bilingual phrases from each alignment, but now many reasonable (and overlapping) possibilities can be included. This transformation function is inspired in recent work done

in phrase-based statistical machine translation. We refer the reader to [16] for details on different methods for extracting bilingual phrases from alignments. It is noteworthy that this is the first time that this statistical algorithm is framed in the GIATI framework. The compound corpus will contain only strings of length one. Let us illustrate this with our example. The alignment above would possibly produce a corpus of phrases with 7 strings of length 1:

> S = {(the, o), (configuration, configuração), (configuration, configuração de), (program, programa), (program, de programa), (configuration program, programa de configuração), (the configuration program, o programa de configuração)}

2. Infer a finite-state automaton: we use a smoothed *unigram* on S with a normalization on the probability of appearance of the input part in each bilingual phrase in S.

3. Undo the transformation: proceed as in algorithm #1.

The basic difference between algorithms #1 and #2 is that the second one produces a transducer with a big amount of translation information, but with a very poor model of how input (and output) words should be concatenated, while the first one includes a smaller amount of translation information (i.e., a smaller amount of possible phrases) but it keeps information about the order in which words appear.

4. A speech-to-speech MT architecture based on composition

The *Audimus* system [13] imposes very few constraints on its search space. The main requirement is that it must consist of a single transducer mapping from acoustic tags (distributions of acoustic features) to output language words. However, in practice, this search space is built as the composition of multiple WFSTs representing various knowledge sources, such as acoustic models ($\mathcal{A}$), lexical models ($\mathcal{W}$), language models ($\mathcal{L}$) or translation models ($\mathcal{T}$). Due to the few assumptions made by the system regarding the search space, other knowledge sources, such as context dependency and pronunciation rules, can be easily integrated as long as they are implemented as transducers.

One advantage of the use of WFSTs, is that the search space can be replaced by an equivalent but more efficient transducer through the use of operations such as weighted determinization, minimization or pushing, see [14].

If we take a look at the size of models in real speech-to-speech tasks, we can readily see that composition of such models as described above would produce huge transducers that would make their practical use unfeasible. Due to this reason, it is essential that the calculation of composition is *not* done by a full expansion of the resulting transducers. *Audimus* uses specialized parsing algorithms so that the generation of the combined transducer is done *on-the-fly* [8]. The combined transducer is expanded on demand while processing the input signal and only the necessary parts (states and transitions) are explicitely represented. These mechanisms are not too time-consuming and they allow to reduce the memory needs to some practical terms.

Table 1: The EuTrans -0 Portuguese-English corpus

Data		Portug.	English
Training	Sentence pairs	10000	
	Running words	89364	96616
	Vocabulary	888	722
Test	Speech utterances	400	—
	Running words	3,700	—

Audimus allows for a very flexible setting, since different models can be combined freely in a composition cascade. The simplest useful possibility is to use a composition of the acoustic, lexical and translation models: $\mathcal{A} \circ \mathcal{W} \circ \mathcal{T}$. Here, the translation model is acting also as a language model for the input and the output languages. However, if we have a good estimation of a language model for the source language, $\mathcal{L}_{in}$ we could introduce it into the translation model as: $\mathcal{A} \circ \mathcal{W} \circ (\mathcal{L}_{in} \circ \mathcal{T})$. Similarly, we could take advantage of a language model for the target language, $\mathcal{L}_{out}$ and calculate $\mathcal{A} \circ \mathcal{W} \circ (\mathcal{T} \circ \mathcal{L}_{out})$. This mechanism has allowed us for the first time to test composition with transducers inferred with GIATI. Results are shown on the next section.

5. Experiments

Audimus has been tested with different translation models on a Portuguese-to-English translation task. A Portuguese-English version of the Spanish-English EuTrans corpus [1] was generated by manual translation of the Spanish portion of the corpus to Portuguese. The EuTrans task is defined on the restricted domain of sentences that a tourist would pronounce at a hotel's desk. It is artificially generated from a set of schemas of sentences. The training corpus has 10,000 sentence pairs.

A multi-speaker Portuguese speech corpus was collected for testing, while no speech was collected for training purposes. The test set consists of 400 utterances: 100 from each of 4 different speakers. The speech was collected using a head-mounted microphone in a typical office environment. A summary of the main features of this Portuguese-English corpus is presented in Table 1.

We have made some experiments to test the perfor-

Table 2: Best TWER results on the Portuguese-English corpus. Algorithms involving $\mathcal{T}_{\#1}$ use 5/3-grams for speech/text input translation models. Those involving $\mathcal{T}_{\#2}$ have used a maximum phrase length of 6 words.

Translation model	speech-input	text-input
$\mathcal{T}_{\#1}$	11.3	8.0
$\mathcal{T}_{\#2}$	67.6	14.5
$\mathcal{L}_{in} \circ \mathcal{T}_{\#1}$	19.6	8.1
$\mathcal{L}_{in} \circ \mathcal{T}_{\#2}$	18.9	14.3
$\mathcal{T}_{\#1} \circ \mathcal{L}_{out}$	23.8	8.1
$\mathcal{T}_{\#2} \circ \mathcal{L}_{out}$	19.1	14.5

mance of *Audimus* given different combinations of language and translation models. The acoustic and lexical models were the same in all cases. Two different translation models were generated from the training corpus using the GIATI algorithms #1 and #2 of Section 3. Let them be $\mathcal{T}_{\#1}$ and $\mathcal{T}_{\#2}$, respectively. Also, a finite-state language model (a smoothed trigram) was generated for the input part of the corpus and another one for the output part. Let them be $\mathcal{L}_{in}$ and $\mathcal{L}_{out}$, respectively. The combinations of transducer and language models that we explored were those shown on Table 2. Our results show that, for this task, algorithm #1 performs better than #2, even if #2 is composed with input or output language models, both for the text and speech tasks.

6. Conclusions

We have explored here the use of a speech recognizer that is able to combine finite-state models through composition, including models that are inferred from parallel corpora by means of different grammatical inference methods. We consider our experimental results to be very promising, specially taking into account the lack of reported results on a speech-to-speech translation task involving European Portuguese.

7. References

[1] J. C. Amengual, J. M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, V. M. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal and J. M. Vilar. *The EUTRANS-I Speech Translation System*, Machine Translation Journal, vol. 15. 2000

[2] S. Bangalore and G. Riccardi, *A Finite-State Approach to Machine Translation*, Proceedings of the North American ACL2001, Pittsburgh, USA. May, 2001.

[3] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer and P. S. Roosin. *A Statistical Approach to Sense Disambiguation in Machine Translation*. Computational Linguistics, 79–86. 1990

[4] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer, *The Mathematics of Statistical Machine Translation: Parameter Estimation*, Computational Linguistics, vol. 19, n. 2, pp. 263–311. 1993

[5] F. Casacuberta, E. Vidal and D. Picó. *Inference of finite-state transducers from regular languages*. To be published in Pattern Recognition.

[6] F. Casacuberta, H. Ney, F.J. Och, E. Vidal, J.M. Vilar, S. Barrachina, I. García-Varea, D. Llorens, C. Martínez, S. Molau, F. Nevado, M. Pastor, D. Picó, A. Sanchis, C. Tillmann. *Some approaches to statistical and finite-state speech-to-speech translation*. Computer Speech and Language, Vol. 18, pp. 25-47, 2004.

[7] F. Casacuberta, E. Vidal, A. Sanchis and J. M. Vilar. *Pattern recognition approaches for speech-to-speech translation*. Cybernetic and Systems: an International Journal, vol.35, n.1, pp. 3–17. 2004.

[8] D. Caseiro, *Finite-State Methods in Automatic Speech Recognition*, Phd Thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa. 2003

[9] J. Eisner, *Parameter Estimation for Probabilistic Finite-State Transducers*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, 2002.

[10] Translation with finite-state devices, K. Knight and Y. Al-Onaizan, Proceedings of the 4th. ANSTA Conference. 1998

[11] D. Llorens, *Suavizado de autómatas y traductores finitos estocásticos*, Phd Thesis, Universitat Politècnica de València. 2000

[12] E. Mäkinen. *Inferring finite transducers*. Technical report A-1999-3, University of Tampere. 1999

[13] H. Meinedo, D. Caseiro, J. Neto and I. Trancoso, *AUDIMUS.media: A Broadcast News Speech Recognition System for the European Portuguese Language*. Lecture Notes in Artificial Intelligence, vol. 2721. Springer-Verlag. 2003

[14] M. Mohri, F. Pereira, M. Riley, *Weighted Finite-State Transducers in Speech Recognition*, Computer Speech and Language, vol. 16, n. 1, pp. 69–88. 2002.

[15] H. Ney, S. Nießen, F. J. Och, H. Sawaf, C. Tillmann, S. Vogel. *Algorithms for statistical translation of spoken language*, IEEE Transactions on Speech and Audio Processing, vol. 8, n. 1, pp. 24–36. 2000

[16] J. Tomás, F. Casacuberta: *Monotone Statistical Translation using Word Groups*. Proceedings of the Machine Translation Summit VIII, Santiago de Compostela, Spain (2001)