GIATI: a general methodology for finite-state translation using alignments

David Picó, Jesús Tomás, and Francisco Casacuberta

Institut Tecnològic d'Informàtica
Universitat Politècnica de València
Valencia, Spain

Summary. Statistical techniques have experienced an increasing interest by the natural language research community in the last years and have proved powerful possibilities for extracting useful information from translation examples. Both statistical language modeling and statistical machine translation are well-established disciplines with solid basis and outstanding results. On the other hand, finite-state transducers have revealed as an efficient and flexible formalism for the representation of a wide range of the kind of information that arises in natural language processing.

This paper presents a powerful general framework for combining statistical techniques with grammatical inference and finite-state traducers. The GIATI methodology proposed here provides a schema for building inference algorithms that are able to generate finite-state transducers from parallel corpora of text making use of information supplied by robust statistical techniques such as n-grams and alignments. Here, the general method is presented together with two concrete inference algorithms and some experiments that show the validness of the GIATI framework for real-world translation tasks.

Key words: Machine translation, finite-state models, statistical alignments

1 Introduction

As it is well known, the fields of statistical and syntactic pattern recognition have found one of their most outstanding applications in natural language processing. Language modeling, machine translation or document retrieval are some examples of interesting tasks that have been successfully approached with pattern recognition techniques and are the object of intensive research. In language modeling, for instance, the statistical technique of smoothed n-grams has become the most widely used solution for problems such as speech recognition, clearly beating other approaches coming from a knowledge-based framework.

Machine translation has been traditionally approached by knowledge-based methods. However, in the last years, encouraging results obtained using statistical methods have raised the claim and the subsequent discussion about the possibility of machine translation to be successfully performed by learning the models from examples, specially in restricted domains of language. Among the pioneer work in statistical machine translation, the commonly known as the "IBM algorithms" [3] were the first to describe the concept of *statistical alignments*, which will center our attention in the following sections, and have produced a wide range of derived techniques that rely to some extent on that seminal work.

The high heterogeneity of the available techniques in these fields is one of the reasons for the increasing interest in formalisms that can establish a common framework for them. Finite-state automata (and the closely related finite-state transducers) are one of such formalisms. They are founded on solid mathematical basis (see, for example, [2]), have many developments in grammatical inference [9], and provide a flexible and efficient tool for representing different kinds of information generated in natural language processing. They have been used in speech recognition [1], morphology, phonotactics, dictionary compression [7], and machine translation [9], among others.

The present work presents a general methodology for representing information coming from different sources (in particular, from statistical alignments) using finite-state transducers, called GIATI (for *grammatical inference and alignments for transducer inference*). This general setting constitutes a a sort of *template* that acts as a melting pot for the creation of algorithms that are able to generate machine translation models (finite-state transducers) starting from bilingual corpora of text. This methodology has been already presented in other forums [4] with an emphasis on the theoretical and algebraic aspects. This paper offers a more concise presentation of GIATI and it focuses on new practical algorithms and experimental results on a real machine translation task about instruction manuals of hardware equipment.

2 The GIATI methodology

2.1 Preliminaries

A *weighted finite-state automata* (WFSA) is a tuple $A = (\Gamma, Q, i, f, P)$, where Γ is an alphabet of symbols, Q is a finite set of states, $i : Q \to \Re$ and $f : Q \to \Re$ give a weight to the possibility of each state to be a initial or final state, respectively, and $P : Q \times \{\Gamma \cup \lambda\} \times Q \to \Re$ defines and gives a weight to each transition between two states, which is labeled with one symbol from Γ or with the empty string, λ. A semi-ring can be defined on the set of weights, so that under some particular conditions (such as the weight values ranging from 0 to 1) the automata defines a probability distribution over the free

monoid $\Gamma^\star$ and is called an *stochastic* finite-state automata. This has been studied in detail in [7].

A *weighted finite-state transducer* (WFST) is defined as a weighted finite-state automata, with the exception that transitions between states are labeled with *pairs* of symbols that belong to a Cartesian product of two different (*input* and *output*) alphabets, $\Sigma \times \Delta$.

A WFSA is essentially a device that can assign a weight to all the strings of symbols that label a path from some input state to some output state. The total weight is usually computed by an accumulated addition or multiplication of the corresponding weights of the transitions in the path. Analogously, a WFST is able to assign weights to *pairs of strings*, following a similar procedure. This opens the door for the possibility of using WFST for translation. If we have an input string and want to find a translation of it, we can perform a search for the set of string pairs (with their corresponding weights) within the transducer such that the input string coincides with the one we want. The standard procedure is to choose the pair with the highest (or lowest) weight and yield the output part as the translation result. This search is not obvious and has been studied in different works (see, for example, [5]).

When an automaton or a transducer are unweighted they behave as accepting machines that accept the strings or string pairs, respectively, that label a path from some input state to some output state. The set of strings accepted by an automaton $\mathcal{A}$ is called the *language* accepted by that automaton, $L(\mathcal{A})$. The set of string pairs accepted by a transducer $\mathcal{T}$ is called the *translation* accepted by that transducer, $T(\mathcal{A})$.

Given two finite alphabets, Γ and Γ', a *morphism* $h : \Gamma^\star \rightarrow \Gamma'^\star$ is a function that satisfies the following conditions: a) $h(\bar{x}, \bar{x}') = h(\bar{x})h(\bar{x}')$, $\forall \bar{x}, \bar{x}' \in \Gamma^\star$, and b) $h(\lambda) = \lambda$, where λ is the empty string. An *alphabetic morphism* h is a morphism that verifies: $h(a) \in \Gamma'$, $\forall a \in \Gamma$. The definition of GIATI in the following subsection makes use of morphisms as a way to denote rewriting transformations between the different set of symbols involved.

2.2 GIATI

The goal of GIATI is to define a general inference method for obtaining a finite-state transducer from a corpus of parallel text[1]. The aim is to produce a transducer that is able to generalize the training data and can find the correct translation of new input sentences that have not been seen during the training process.

The process defined by GIATI is illustrated in Figure 1. Given a parallel corpus consisting in a finite sample A of string pairs $(\bar{s}, \bar{t}) \in \Sigma^\star \times \Delta^\star$:

[1] A corpus of parallel text, also called a parallel corpus or a *bicorpus*, is a collection of pairs of word strings, usually sentences in the linguistic sense, that belong to two different languages and are the translation from one another.

1. Each training pair $(\bar{s}, \bar{t})$ from A is transformed into a string z from an *extended alphabet* Γ yielding a sample S of strings, $S \subset \Gamma^\star$.
2. A (stochastic) finite-state transducer $\mathcal{A}$ is inferred from S.
3. The symbols (from Γ) of edges in $\mathcal{A}$ are transformed back into pairs of strings of source/target symbols (from $\Sigma^\star \times \Delta^\star$).

The first transformation is modeled by some labeling function $\mathcal{L} : \Sigma^\star \times \Delta^\star \rightarrow \Gamma^\star$, while the last transformation is defined by an "inverse labeling function" $\Lambda(\cdot)$, such that $\Lambda(\mathcal{L}(A)) = A$. Typically, $\Lambda(\cdot)$ consists of a couple of morphisms, h_Σ, h_Δ, such that for any string $z \in \Gamma^\star$, $\Lambda(z) = (h_\Sigma(z), h_\Delta(z))$. This guarantees some conditions on the transformations which are helpful to demonstrate some interesting algebraic properties (see [4]).

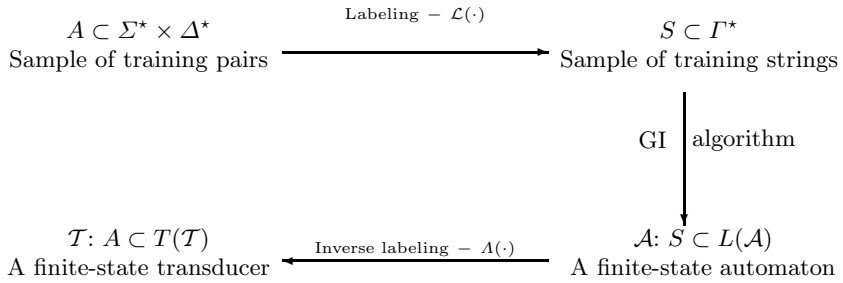

Fig. 1. Basic scheme for the inference of finite-state transducers. A is a finite sample of training pairs. S is a finite sample of strings. $\mathcal{A}$ is an automaton inferred from S such that S is a subset of the language $L(G)$. $\mathcal{T}$ is a finite-state transducer whose translation $T(\mathcal{T})$ includes the training sample A.

The interest of GIATI comes from the fact that it establishes a general template the instances of which are different inference algorithms. So, for example, the first transformation, $\mathcal{L}$, can aim at grouping pairs of segments of words into bigger, meaningful units that can be considered as the new tokens for the sample of training strings S. In the practical applications that we are presenting in Section 4 for machine translation this is done in different manners but taking profit of the information given by statistical alignments of words (see the next section). The grammatical inference algorithm that is needed in the second step can be any algorithm able to infer a WFSA from a corpus of text. For example, different variants of smoothed n-grams can be used for this [6].

A toy example: inferring a canonical transducer

In this example, the algorithm produced by GIATI infers an unweighted finite-state transducer that will only accept the source sentences of the training set and will produce the target sentences as the corresponding translation.

- Transformation of string pairs into strings: Given a training pair $(\bar{s}, \bar{t})$, each symbol of $\bar{s}$ is labeled as itself, except for the last one, x, which is labeled as $x_{\bar{s}}$, i.e., a new symbol composed by the symbol itself and the target sentence as subindex:
  $\Gamma = \{a, b, a\_\{00\}, a\_\{101\}, a\_\{011\}, a\_\{0\}\}$
  $S = \{(abba\_\{00\}), (aaabbaa\_\{101\}), (bbaaa\_\{011\}), (bba\_\{0\})\}$

- Inference of a finite-state automaton: We will use a prefix tree-acceptor. The automaton produced by this method is shown in Fig. 2.

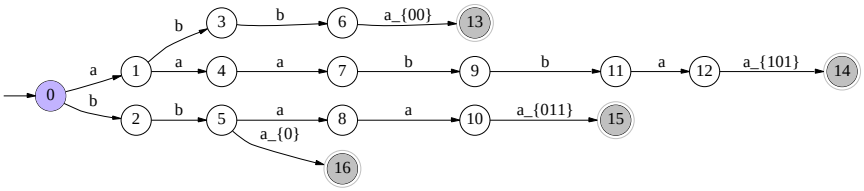

Fig. 2. A prefix–tree acceptor for the training sample of Example 1.

- Inverse transformation:

	a	b	a_{00}	a_{101}	a_{011}	a_0
h_Σ	a	b	a	a	a	a
h_Δ	λ	λ	00	101	011	0

  The resulting canonical finite-state transducer is shown in Fig. 3.

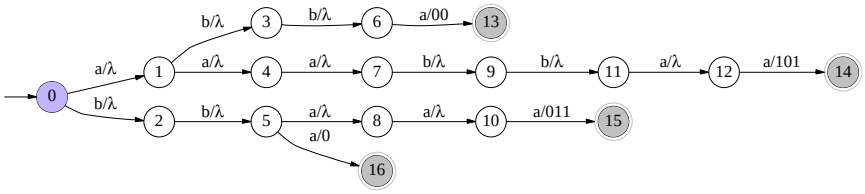

Fig. 3. The resulting finite-state transducer for Example 1.

3 Statistical alignments

Our aim when building a combined corpus is to condense meaningful information about the relations that lay between the input and output words. This 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. The interesting thing is the availibily of algorithms for learning such correspondences from bilingual corpora. 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.

Formally, an alignment between a pair of sentences $(\bar{s}, \bar{t})$ is a mapping $i \rightarrow j = a_j$ that assigns a word s_j in position j to a word t_i in position $i = a_j$. Alignments are used as a hidden variable in statistical machine translation models such as IBM models [3] or hidden Markov models [10].

4 Two translation algorithms

In order to explain the following algorithms clearly we will denote a pair from the parallel corpus by $(\bar{s}, \bar{t})$ and we will be using a very small example of alignment taken from a real English-to-Spanish corpus: We will consider that the English phrase *the configuration program* is aligned with the Spanish phrase *el programa de configuración* with the alignment $\{1 \rightarrow 1, 2 \rightarrow 4, 3 \rightarrow 3\}$.

Algorithm #1: using a language of segment pairs

- Transformation of string pairs into strings: the composed string is a sequence of $|\bar{s}|$ pairs, $(u_i, \bar{v}_i)$, where $u_i = s_i$ and $\bar{v}_1 \bar{v}_2 \ldots \bar{v}_{|\bar{s}|} = \bar{t}$. Each of these pairs is considered to be *a single symbol*. We refer the reader to [4] for a complete description of this algorithm and other minor details. Applying this algorithm to the alignment of our example would produce the following corpus containing one string:
  $S = \{$(the, el) (configuration, λ), (program, programa de configuración)$\}$
- Inference of 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 [6].
- Inverse transformation: the transitions in the inferred automaton are labeled with compound symbols which are pairs of strings. A transducer can be obtained directly by considering these symbols as the pair of strings that label a transducer transition.

Algorithm #2: using a corpus of bilingual phrases

- Transformation of string pairs into strings: this transformation obtains a set of bilingual phrases from each alignment, where many reasonable (and

overlapping) possibilities are included. The compound corpus of strings only contains strings of length one and the symbols are pairs of strings as in the previous algorithm. Let us illustrate this with our small example. The alignment above will produce a corpus of phrases such as the following one, containing 7 strings of length 1:

$S = \{$(the, el), (configuration, configuración), (configuration, configuración de), (program, programa), (program, de programa), (configuration program, programa de configuración), (the configuration program, el programa de configuración)$\}$

This transformation function is inspired in recent work done in phrase-based statistical machine translation. We refer the reader to [10, 8] for details on different methods for extracting bilingual phrases from alignments.

- Inference of 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.
- Inverse transformation: the same as in algorithm #1.

5 Experimental results

We have performed some experiments in a real-word machine translation task so as to test the feasibility of the algorithms explained in the previous section. This corpus consists of a collection of technical manuals of hardware equipment by the Xerox company and have been processed by the team of the TransType2 project [11]. We are using the English-to-Spanish version of the corpus. The characteristics of this corpus are shown in Figure 5.

Training set	English	Spanish
Number of sentences	56,773	56,773
Running words	679,678	768,564
Size of the vocabulary	7,976	11,094
Test set	English	Spanish
Number of sentences	1,125	1,125
Running words	10,106	8,370
Size of the vocabulary	1,132	1,215

Fig. 4. Size values of the TT2 training and test corpora.

We have used the measure known as *word error rate* (WER), calculated as the percentage of insertions, deletions and substitutions of words that are necessary to obtain the reference output sentence from the translation calculated by the algorithm. Our results are a WER of 34.0% for the algorithm #1 using 4-grams smoothed by back-off, and 30.8% for the algorithm #2 limiting

the length of phrases to 6 words. These experiments are still in a rudimentary stage but the results are not in very different range of error that those obtained by other more sofisticated methods. For example, phrase-based statistical translation with monotone search [8] obtained for this task a WER of 24.87% using much longer phrases and lexical weighting.

6 Conclusions and further work

The GIATI methodology for inferring finite-state transducers from parallel corpora has been presented here. GIATI is a general way of defining transducer inference algorithms making use of automata induction and other kinds of useful information such as statistical alignments. Two inference algorithms described within the GIATI framework have been presented here and some encouraging experimental results have been reported.

Further work on GIATI points towards searching other algorithms that make a clever use of statistical alignments in combination with n-grams or other finite-state automata inference methods. GIATI and finite-state models seem to be specially well suited for highly sequential translation tasks. Reordering words is usually a problem. We want to explore the possibilities for incorporating non-monotonous information (recursive alignments, statistical non-monotonous search) through some extension of GIATI.

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. J. Berstel, *Transductions and context-free languages*, Teubner Stuttgart. 1979
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. F. Casacuberta, E. Vidal and D. Picó. *Inference of finite-state transducers from regular languages.* Acepted for publications in Pattern Recognition.
5. F. Casacuberta and C. de la Higuera, *Linguistic Decoding is a Difficult Computational Problem*, Pattern Recognition Letters, vol. 20. 1999
6. D. Llorens, *Suavizado de autómatas y traductores finitos estocásticos*, Phd Thesis, Universitat Politècnica de València. 2000
7. M. Mohri, *Finite-State Transducers in Language and Speech Processing*, Computational Linguistics, vol. 23. 1997
8. J. Tomás, Casacuberta, F.: *Monotone Statistical Translation using Word Groups.* Proceedings of the Machine Translation Summit VIII, Santiago de Compostela, Spain (2001)

9. J. M. Vilar, *Improve the Learning of subsequential Transducers by Using Alignments and Dictionaries*, Grammatical Inference: Algorithms and Applications, Springer-Verlag, vol. 1891. 2000
10. R. Zens, F. J. Och and H. Ney. *Phrase-based statistical machine translation*. `http://citeseer.nj.nec.com/zens02phrasebased.html`
11. TransType 2 Project. `http://tt2.sema.es`