Learning Extended Finite State Models for Language Translation[1]

J. M. Vilar,[2] E. Vidal and
Dpto. Sistemas Informáticos y Computación
Universidad Politécnica de Valencia,
46020 Valencia, SPAIN.
{jvilar,evidal}@iti.upv.es

J. C. Amengual
Unidad Predepartamental de Informática
Universidad Jaume I,
12071 Castellón, SPAIN.
jcamen@inf.uji.es

Abstract. The use of Subsequential Transducers (a kind of Finite-State Models) in Automatic Translation applications is considered. A methodology that improves the performance of the learning algorithm by means of an automatic reordering of the output sentences is presented. This technique yields a greater degree of synchrony between the input and output samples. The proposed approach leads to a reduction in the number of samples necessary to learn the transducer and a reduction in the size of the model so obtained.

1 Introduction

Recently, the use of Finite-State (FS) models has been proposed for Language Translation (LT) applications. While this kind of models is often considered too simplistic to properly approach such a complex problem, results show that they can perform surprisingly well in Limited Domain (LD) tasks; that is, tasks with small or medium sized vocabulary and restricted semantic scope [7, 9, 12]. One of the reasons for this success lies on the fact that although natural languages are complex, the mappings defined by their translations can be comparatively much simpler, specially when these languages are close as is the case with many European languages.

Among the many attractive features of FS models for LT, an important one is the ease with which these models can be *tightly integrated* with standard acoustic-phonetic models of the input language, readily yielding quite effective *speech-input* LT systems [9]. Thanks to their conceptual and structural simplicity, these systems are significantly more robust than others based on the more conventional approach of loosely coupling an existing LT package to the output of a speech recognition front-end.

In this paper we focus on *Subsequential Transducers* (SSTs) [4]. Output symbols or substrings are generated by a SST only after having seen enough input symbols to guarantee a correct output. The amount of symbols to wait for may be variable and context-dependent and it may also be necessary to produce output after the whole input has been seen. This allows for larger "asynchrony" between the input and the output sentences than with other simpler FS models such as Sequential Transducers and Mealy or Moore machines [4]. It should be noted that many translation tasks that may appear much more difficult, are inherently of this *subsequential* nature. For instance, we can always translate natural English sentences into correct Spanish by successively outputting Spanish words that can be determined from a finite (often short) sequence of previously seen English words. In other words, we do not need to wait for a whole discourse to end before starting the translation.

A distinctive advantage of SSTs is that they can be learned in a completely automatic manner from a sufficiently large corpus of training data by using a recently proposed algorithm [10, 11]. Using this algorithm, a number of experiments have been carried out so far with LD LT, including speech-input applications [7, 9, 12, 14]. These works aimed at solving different problems arising in this kind of application. In particular, the need of Input/Output *Language Models* to cope with the distortions and noise involved by speech-input operation was first considered in [12] and specific techniques to keep the required amount of training data at reasonably small levels were first studied in [14]. In this paper we go deeper into the latter of the above issues and propose new techniques to assist the basic SST learning algorithm by reducing the effective Input/Output asynchrony that has to be modeled by the learned devices.

2 Subsequential Transducer Learning: Basic Concepts and Previous Work

A subsequential transducer is a deterministic finite-state network that accepts sentences from a given input language and produces associated sentences of an output language. Each edge of the network has associated an input symbol and an output string. Every time an input symbol is accepted, the corresponding string is output and a new state is reached. After the whole input is processed, additional output may be produced from the last state reached in the analysis of the input [4].

Given a set of training pairs of sentences from a translation task, the *Onward Subsequential Transducer Inference Algorithm* (OSTIA) learns a SST that generalizes the training set [10, 11]. The algorithm builds a straightforward prefix-

[1] Work partially supported by the Spanish CICYT under grant TIC95–0984–C02–01

[2] Supported by a grant of the Spanish Ministerio de Educación y Ciencia

Proceedings of the ECAI 96 Workshop
Extended Finite State Models of Language
Edited by A. Kornai.

tree representation of all the training pairs and moves the output strings towards the root of this tree as much as possible, leading to an *"onward"* tree representation. Finally a state merging process is carried out. The algorithm guarantees identification of the target transduction in the limit; that is, if the unknown target translation exhibits a subsequential structure, convergence to it is guaranteed whenever the set of training samples is representative [10, 11].

Additionally, if models for the input and/or output languages are available, an extended version of OSTIA can be used which produces SSTs that only accept input sentences and only produce output sentences compatible with these models [9, 12]. This becomes of paramount importance when noisy and distorted input like speech is expected.

SSTs base their translation ability on "delaying" the production of output words until enough of the input sentence has been seen to guarantee a correct output. This is illustrated in the following example of Spanish/English translation (from Feldman's task [8, 7]):

```
se añade un triángulo grande y claro .
  a large light triangle is added .
```

The input Spanish sentence is translated into English by following a sequence of states in a SST such that the input words "se añade" produce no output (though they change the state of the SST), the word "un" produces "a" as output, the words "triángulo", "grande" and "y" do not produce any output string (though they change the state), the word "claro" yields "large light triangle" and the end-of-sentence period produces "is added .".

Every word sequence whose translation must be delayed is "stored" by means of the states of the SST. While OSTIA has proved both theoretically and practically able to learn (possibly large) SSTs that can cope with usual Input/Output asynchronies, when the number of (functionally equivalent) words increases, the required number of states can grow as much as $O(n^k)$, where n is the number of words and k the required delay. Clearly, for realistic tasks, the amount of training data required to help learning all the possible combinations could go far beyond practical limits.

In [14] a first approach was proposed to tackle one part of this problem; namely, the growth with the number of words n. The basic idea was to rely on *word clusters* rather than actual words. With the help of a dictionary, words can be grouped into "paired clusters" (in both languages) for learning. Also, in the test phase, the dictionary is used to recover the actual identity of the words within each output cluster. This preserves the essential FS nature of the translation model, while drastically cutting down the size of the learned models and the corresponding demand of training data. Experiments reported in [14] showed that the vocabulary can be increased from about 40 words to more than 300 without significant performance degradation or increase in the amount of training data required.

Nevertheless, the above mentioned exponential growth can still become prohibitive because of the exponent k; that is, if the translation of even a small number of words or word clusters need to be delayed a long extend. In this work we propose new techniques to tackle this second part of the problem, while essentially keeping the very convenient finite-state nature of the translation model. The aim is to reduce the degree of Input/Output asynchrony that has to be modeled by a SST which is learned for a given task.

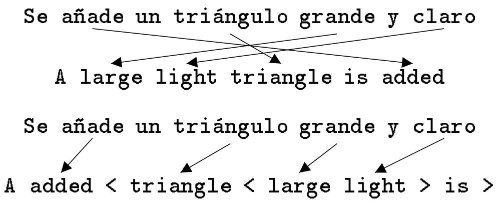
Se añade un triángulo grande y claro
A large light triangle is added
Se añade un triángulo grande y claro
A added < triangle < large light > is >

Figure 1. An example of partial alignment and reordering. Original pair and a partial alignment (above). Resulting pair after the reordering (below).

3 Coping with Input/Output Asynchrony and Lexicon Size

The techniques introduced here rely on using *aligned* pairs of training sentences. Such kind of data can be found in certain open domain corpora, but for limited domain applications unaligned pairs of sentences are often the only data available. Nonetheless, rough, partial alignments can be easily obtained by pairing input/output words of each training sentence with the help of a (probabilistic) dictionary. Techniques to automatically build bilingual dictionaries from training parallel text have been proposed recently. In this work we use the so called "IBM Model-1" [6]. This simple stochastic translation model can be optimally trained from paired sentences and produces, as a byproduct, a stochastic dictionary. Those pairs of words having high likelihood of being translation of each-other are used to obtain the required partial alignments. This is illustrated in the example of Figure 1 (above).

Since only high probability input/output relations are used, these partial alignments can be considered robust enough to be used as "anchor associations" to reorder the words of the output sentences of the training pairs so that, hopefully, the most prominent long-term asynchronies are removed. Once a training set of sentences have been processed in this way, the OSTI algorithm can be used to learn a SST that accounts for the mapping from the input language to a "reordered version" of the output language. This mapping is expected to be much simpler than the original one in the sense that much less delay will be required to produce the (reordered) output tokens. Obviously, the reordering mechanism must provide adequate means to recover a correct output order for each input (test) sentence. To this end we use pairs of *brackets* to adequately mark the reordered training output sentences, as shown in Figure 1. From these marks, the original order can be easily recovered by placing each word preceding a left bracket after the corresponding (paired) right bracket.

Given a training set S of pairs of input/output sentences (x, y), the proposed training approach can be summarized as follows:

1. Train IBM Model-1 on S and obtain a probabilistic dictionary D.
2. Prune from D those pairs of words with probability below a threshold.

3. Partially align the pairs of sentences in S using the pruned D.
4. Reorder and bracket the output sentences of S to produce S'.
5. Using OSTIA, learn a SST T from S'.

Now, given a new test input sentence x the system produces a translation y through the following three simple steps:

1. Obtain the translation y' of x through T
2. Reorder y' with the help of its embedded brackets.
3. Remove these brackets to obtain y.

4 The Reordering Algorithm

The reordering is done by scanning the output sentence from left to right and creating a new sentence s along the way. If the word under consideration is not aligned nor crosses with any other, it is appended to s. In case there is a crossing, s is examined right to left, skipping those parts already bracketed, until a position is found such that it has no crossings. The word is inserted in that position, together with an opening bracket and a closing bracket is appended to s.

The process can be seen with the help of the sentence from the example in Figure 1:

Step	Word	Result (s)
1	A	A
2	large	A large
3	light	A large light
4	triangle	A triangle ¡ large light ¿
5	is	A triangle ¡ large light ¿ is
6	added	A added ¡ triangle ¡ large light ¿ is ¿

No reordering is necessary in the first three steps. A crossing appears when "`triangle`" is examined, it is then inserted before "`large`" to avoid the crossing. The next word is appended without trouble and finally "`added`" is placed just before "`triangle`". Note that when searching the placement of "`added`" the words "`large`" and "`light`" are not examined since they are already bracketed.

5 Output Language Modeling: Balancing the Brackets in the SST Translation of Test Sentences

One possible problem with this reordering of the output is that there is no guarantee that the transducers learned by OSTIA perfectly generalize the bracketing of the training sentences so that brackets will be balanced for all possible test sentences. This becomes even more problematic when noisy input is considered, as is the case in speech-input applications. Given the finite state nature of the SST, keeping the balance is of course impossible in general, but in practice simple solutions can be used if a maximum depth of the brackets is fixed. If k is the highest level of bracketing, balance can be achieved by using a simple output "Language Model" consisting in an automaton built as follows:

1. Create one state for each of the levels $0 \dots k$. Consider the state numbered 0 both as the initial and the only final one.
2. For each symbol different from a bracket create a transition from each state to itself, labeling the transition with that symbol.
3. For each level l, connect the state at level l with that at level $l+1$ with an arc labeled with an opening bracket and with that at level $l-1$ with an arc labeled with a closing bracket.

The result of these steps is an automaton that only accepts sentences correctly bracketed. By using it as a model of the output language for OSTIA the desired effect is achieved. In case a better language model is available, (which in general may not enforce the bracketing), the standard construction for the automaton corresponding to the intersection of two languages can be used.

6 Coping with Noisy Input

In practice, the performance of the SST models tends to degrade dramatically when the input sentences do not strictly comply with the linguistic restrictions imposed by the input language model. This problem can be solved to some extent by means of Error-Correcting Parsing [1, 2, 3].

Under this approach, the input sentence, x, is considered as a corrupted version of some sentence $\hat{x} \in L$ (L being the domain of the SST). The corruption process is modelled by means of an Error Model E, that comprises insertions, substitutions and deletions. The parsing of an input sentence x consists then in finding the corresponding string in L which has a maximum posterior probability; that is,

$$\hat{x} = \mathrm{argmax}_{x' \in L} P_L(x') P_E(x|x'),$$

where $P_L(x')$ is the probability of x' with regard to L, given by the (input part of the) SST, and $P_E(x|x')$ is the probability of x being a corrupted version of x' according to E. Finally, the translation of $\hat{x}$ through the SST, y', is the one actually used in the reordering step for obtaining the final translation.

The above mentioned probabilities can be trained using a corpus S' which is a distorted version of S (the training corpus). An initial model is constructed on the base of the Levenshtein distance between corrupted and clean sentences. Similarly, $P_L(\cdot)$ is initialized using a grammar of L with uniform distribution of edge probabilities. Then an stochastic error-correcting parsing of S' yields new estimates of $P_L(\cdot)$ and $P_E(\cdot|\cdot)$ and this parsing and estimation process can be iterated until convergence.

7 Experiments

Spanish-English translation experiments were carried out with an extension of the so-called Miniature Language Acquisition Task recently proposed by Feldman et al. [7, 8]. A set of 16000 pairs was used for training, and a separate set of 10000 Spanish sentences was used for testing. SSTs were learned from the training set using both the direct approach and the above described reordering scheme. In both cases, a 4-Gram language model for the domain (Spanish), learned from the input sentences of the training pairs, was used for learning the SSTs. Finally, the probabilities of the resulting SSTs were estimated from the same training data.

Table 1. Experimental results for the Feldman's Task. Left: word error rates for distorted input (5% word error rate). In brackets, model sizes (states/edges).

Train. size	Direct	Reordered
1,000	44.0% (813 / 2023)	17.6% (532 / 1338)
2,000	37.8% (1406 / 3353)	6.2% (358 / 979)
4,000	25.2% (1686 / 4051)	2.2% (144 / 440)
8,000	2.7% (244 / 719)	1.7% (109 / 344)
16,000	1.7% (100 / 363)	1.7% (63 / 183)

In order to approach the conditions of speech input operation the test sentences were distorted through a noisy channel, C, involving (equally probable) word insertions, deletions and substitutions. The overall distortion (word error rate) was 5%, some examples of distorted sentences can be seen in Figure 2 on p. 96. Testing was carried out through stochastic error-correcting parsing with the corresponding probabilities estimated from a distorted version of the training data [1, 2, 3]. The results, shown in Table 1, confirm that the rate of learning is significantly higher using the proposed reordering scheme and the obtained models are smaller.

It should be noted that the sizes of the learned models tend to decrease with the growth of the amount of training data. In the limit, as the training set becomes completely representative of a source *subsequential transduction*, the learning algorithm is guaranteed to yield a canonical (minimum-size) subsequential transducer [11, 12]. Thus the progress of learning generally entails a reduction of both the model size and the error rate.

8 Discussion

A new technique that helps in mitigating one of the difficulties in learning (finite-state) translation models has been presented. Although it has been tested on a particular algorithm (OSTIA) its application is general since it only modifies the presentation of the data. This technique relies on the ability to find aligments on the training data, although they need not be complete. A limitation of the technique is that it only allows the movement of single words, but we are working in a version that considers the reordering of whole groups of words.

REFERENCES

[1] J.C. Amengual, E. Vidal. "Canonización del Lenguaje mediante Técnicas de Corrección de Errores" (in Spanish). Technical Report, DSIC-II/17/95. Depto. de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. Spain. September, 1995.

[2] J.C. Amengual, E. Vidal and J.M. Benedí. "Simplifying Language through Error-Correcting Decoding". Proceedings of the ICSLP96. To be published. 1996.

[3] L. Baahl and F. Jelinek. "Decoding for Channels with Insertions, Deletions and Substitutions with Applications to Speech Recognition". *IEEE Transactions on Information Theory.* Vol.IT-21, No.4, pp.404-411. July, 1975.

[4] J. Berstel. *Transductions and Context-Free Languages.* Teubner, Stuttgart. 1979.

[5] P.F. Brown et al.. "A Statistical Approach to Machine Translation". *Computational Linguistics*, Vol. 16, No.2, pp.79-85, 1990.

[6] 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, No.2, pp.263-311, 1993.

[7] A. Castellanos, E. Vidal, I. Galiano. "Application of OSTIA To Machine Translation Tasks". *2nd International Colloquium on Grammatical Inference*, proc., Alicante, Spain, Sept., 1994.

[8] J.A. Feldman, G. Lakoff, A. Stolcke, S.H. Weber. "Miniature Language Acquisition: A touchstone for cognitive science". Technical Report, TR-90-009. ICSI, Berkeley, California. April, 1990.

[9] V.M. Jiménez, A. Castellanos, E. Vidal, J. Oncina "Some Results with a Trainable Speech Translation and Understanding System". Proc. of ICASSP95, pp. 113-116. 1995.

[10] J. Oncina. "Aprendizaje de Lenguages Regulares y Funciones Subsecuenciales". Ph.D. diss., Universidad Politécnica de Valencia, 1991.

[11] J. Oncina, P. García, E. Vidal. "Learning Subsequential Transducers for Pattern Recognition Interpretation Tasks". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.15, No.5, pp.448-458. May, 1993.

[12] J. Oncina, A. Castellanos, E. Vidal, V. Jiménez. "Corpus-Based Machine Translation through Subsequential Transducers". *Third Int. Conf. on the Cognitive Science of Natural Language Processing*, proc., Dublin, 1994

[13] E. Vidal, F. Casacuberta, P. García. "Grammatical Inference and Automatic Speech Recognition". *In Speech Recognition and Coding. New Advances and Trends*, J. Rubio and J.M. López, Eds. Springer Verlag, 1994.

[14] J.M. Vilar, A. Marzal, E. Vidal. "Learning Language Translation in Limited Domains using Finite-State Models: some Extensions and Improvements". EUROSPEECH-95, Proc. Madrid, 1995.

Original:	se elimina el circulo grande y claro que esta muy por encima del triangulo claro y del triangulo mediano y claro
Distorted:	se elimina y el circulo grande y claro esta muy por encima triangulo claro y del triangulo mediano un claro
Translation:	the large light circle which is far above the light triangle and the medium light triangle is removed
Original:	un circulo mediano y claro esta debajo de un cuadrado pequeNo y claro y un triangulo pequeNo y oscuro
Distorted:	un tocan circulo mediano y claro esta de un cuadrado pequeNo claro y un triangulo pequeNo y oscuro
Translation:	a medium light circle is below a small light square and a small dark triangle

Figure 2. Some examples of original and 5%-distorted MLA Spanish sentences, together with the corresponding English translations.