Finite-State Models for Machine Translation

Enrique Vidal & Francisco Casacuberta

Instituto Tecnológico de Informática
&
Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

ICGI-2004

Brief introduction

• 10+ years of research on learning Stochastic Finite State Transducer (SFST), mainly for Machine Translation (MT)

• Initial approaches: “pure” GI-based

• Several projects funded by National and European agencies, as well as by private companies

• Always competing with other approaches: knowledge-based, memory-based and “pure” statistical techniques

• Ever increasing demands: larger vocabularies, more open domains, more difficult pairs of languages, less relative amounts of training data

This talk is all about the struggle to evolve our initial, GI-based, SFST learning technology in order to cope with these increasing demands
Index

1 Probabilistic statement of MT problems ▶ 3
2 Rational or finite-state transduction (FST) ▶ 9
3 Subsequential Transduction: “OSTI” algorithm ▶ 15
4 Using input/output syntactic constraints: OSTIA-DR ▶ 30
5 OSTIA-DR: improving scalability ▶ 37
6 Statistical alignments: hybrid methods ▶ 45
7 Alignment-controlled state merging: OMEGA ▶ 48
8 Alignments and bilingual segmentation: GIATI ▶ 52
9 Speech-to-speech and computer assisted translation ▶ 61
10 GIATI revisited: pure statistical learning ▶ 66
**Probabilistic formulation: Text-to-text MT**

Given a source text $x$, its most probable translation is given by:

$$\hat{y} = \arg\max_y \Pr(y \mid x)$$

**Two paradigms:**

- **Statistical alignment approach** [Brown et al., 93]:
  $$\hat{y} = \arg\max_y \Pr(y) \cdot \Pr(x \mid y)$$
  $\Pr(y) \to$ target *language model* (e.g., N-Gram),
  $\Pr(x \mid y) \to$ *stochastic dictionaries* plus *alignment models*.

- **Finite-state approach**
  [Vidal, 97], [Casacuberta & Vidal, Comp. Linguistics 04]:
  $$\hat{y} = \arg\max_y \Pr(x, y)$$
  $\Pr(x, y) \to$ *stochastic finite-state transducer*

**Statistical formulation: Speech-input MT (SMT)**

Given a source utterance (acoustic sequence) $v$, find a target language sentence $\hat{y}$:

$$\hat{y} = \arg\max_y \Pr(y \mid v)$$

This can be broken down into two processes:

$$v \longrightarrow x \longrightarrow y$$

The *hidden variable* $x$ is any possible decoding of $v$, and $y$ is the translation of $x$:

$$\hat{y} = \arg\max_y \sum_x \Pr(y, x \mid v) = \arg\max_y \sum_x \Pr(x, y) \cdot \Pr(v \mid x)$$

$$\hat{y} \approx \arg\max_y \max_x \Pr(x, y) \cdot \Pr(v \mid x)$$

$\Pr(x, y) \to$ *stochastic finite-state transducer*, $\Pr(v \mid x) \to$ *hidden Markov models*
Given a source text $x$ and a fixed prefix of the target sentence $y_p$ (previously validated by the human translator), find a suffix of the target sentence $\hat{y}_s$:

$$\hat{y}_s = \arg\max_{y_s} \Pr(y_s|x, y_p)$$

Taking into account that $\Pr(y_p|x)$ does not depend on $y_s$:

$$\hat{y}_s = \arg\max_{y_s} \Pr(x, y_p y_s)$$

where $y_p y_s$ is the concatenation of the given prefix $y_p$ and a system suggested suffix $y_s$.

$\Pr(x, y_p y_s) \rightarrow$ stochastic finite-state transducer

All the above formulations share the common learning problem of estimating $\Pr(x, y)$; that is, training a SFST from a parallel corpus of source-target sentences.
Not all the transduction tasks are equally difficult

1. Spanish to English, word by word
2. Division by 7
3. English to Decimal
4. Roman to Decimal
5. ATIS: English to "Pseudo English"
6. Spanish to English

The main concern is the required degree of "sequentiality" or position monotonicity between input-output subsequences.
Finite state transducers (FST)

A finite state or rational transducer $\tau$ is a 6-tuple $\tau = (Q, X, Y, q_0, Q_F, E)$:

- $Q$: Finite set of states
- $X, Y$: Input and output alphabets
- $q_0 \in Q$: Initial state
- $Q_F \subset Q$: Set of final states
- $E \subset Q \times X^* \times Y^* \times Q$: "Edges" or transitions

Transitions can be equivalently defined as $E \subset Q \times (X \cup \lambda) \times Y^* \times Q$.

**Example**

$$T_\tau = \{ (\lambda, \lambda), (cb, 213), (ccb, 2213), (a, \lambda), (ac, 003), (cac, 2003), (c, 2), (bc, 111), (bcc, 2111), (b, 13), (bc, 113), (bcc, 213003), (ca, 2), (bc, 13003), (bcc, 1113), (cc, 22), (cca, 22), \ldots \ldots \}$$

Three possible types of ambiguity: input, output and path

Stochastic finite state transducers (SFST)

A stochastic FST (SFST) $T$ is defined by $(\tau, P, P_F)$, where:

- $\tau = (Q, X, Y, q_0, Q_F, E)$ is a FST
- $P: E \to \mathbb{R}^+$ and $P_F: Q_F \to \mathbb{R}^+$ are functions such that:
  \[
  \sum_{(q', u, v, q) \in E} P(q', u, v, q) + P_F(q') = 1 \quad \forall q' \in Q
  \]

- A path $P$ of $T$ is a sequence of transitions of $E$ which starts in $q_0$ and ends in a final state. $P(x, y)$ denotes the set of paths in $T$ matching $x, y$.

- Probability of a path, $P_f$, ending at a final state $q_f$:
  \[
  Pr(P_f) = \prod_{(q', u, v, q) \in P_f} P(q', u, v, q) P_F(q_f)
  \]

- Probability of a translation $(x, y)$ given by $T$:
  \[
  P_T(x, y) = \sum_{P_f \in P(x, y)} Pr(P_f) = \sum_{P_f \in P(x, y)} \prod_{(q', u, v, q) \in P_f} P(q', u, v, q) P_F(q_f)
  \]
  $P_T(x, y)$ defines a joint distribution in $X^*, Y^*$

October 2004

E. Vidal, F. Casacuberta – ITI-UPV-DSIC

FSMT: 10
Stochastic finite state transducers: search problems

- **Most probable translation:** given \( x \in X^* \), find

\[
\hat{y} = \arg\max_{y \in Y^*} P_T(x, y)
\]

*No efficient solution known* (shown to be NP-Hard! [Casacuberta & De la Higuera, 2000]).

**Approximation:**

\[
\hat{y} = \arg\max_{P \in \mathcal{P}} P_r(P)
\]

*Efficient solution by Viterbi search*

- **Most probable path:** given \( T, x \in X^*, y \in Y^* \), find

\[
\hat{P} = \arg\max_{P \in \mathcal{P}} P_r(P)
\]

*Efficient solution by dynamic programming*

*Both problems are easy if \( \tau \) is un-ambiguous – trivial if \( \tau \) is deterministic*

Learning stochastic finite state transducers

Three main families of techniques to learn a SFST from a parallel corpus of source-target sentences:

- **Traditional syntactic pattern recognition paradigm:**
  - Learn the SFST “topology” (the states and transitions)
  - Estimate the probabilities from the same data

*Problem:* The class of finite-state transducers as a whole is at least as hard to learn as the class of finite-state automata!

\( \Rightarrow \) Try to learn adequate subclasses and/or use heuristics!

- **Hybrid methods:** Under the traditional paradigm, use statistical methods to guide the structure learning

- **Pure statistical approach** *(new):*
  - Adequately parameterize the SFST structure and consider it as a hidden variable
  - Estimate everything by Expectation Maximization (EM)
Estimating probabilities of stochastic finite state transducers

- **Estimating transition and final-state probabilities:**
  - **Un-ambiguous transducers:**
    Maximum likelihood estimation from the frequency of use of transition and states in the paths matching the training pairs
  - **Ambiguous transducers:**
    EM re-estimation based on a forward-backward-like algorithm or a Viterbi-like approximation [Picó & Casacuberta, 01]

- **Modeling of unseen events – smoothing:**
  - **Stochastic error-correcting parsing**
    Given a source sentence, $x$, find a path in the transducer that error-correcting matches $x$ with maximum probability
  - **Back-off and interpolation**
    Adapted from techniques used in language modeling [Llorens 01] (so far fully developed only for techniques based on N-Grams)
Not all the transduction tasks are equally difficult

1. Spanish to English, word by word
2. Division by 7
3. English to Decimal
4. Roman to Decimal
5. ATIS: English to "Pseudo English"
6. Spanish to English

The main concern is the required degree of “sequentiality” or position monotonicity between input-output subsequences.

Subsequential transduction

A Subsequential transducer (SST) \( \tau \) is a 6-tuple \( \tau = (Q, X, Y, q_0, E, \sigma) \), where:

- \( Q \) is a finite set of states
- \( X, Y \) are the input and output alphabets
- \( q_0 \in Q \) is the initial state
- \( E \subseteq Q \times X \times Y^* \times Q \) are the “edges” or transitions
- \( \sigma : Q \to Y^* \) is a state output (partial) function

\( \triangleright \) States with \( \sigma(q) \neq \emptyset \) are accepting
\( \triangleright \) Edges are deterministic: \( (q, a, u, r), (q, a, v, s) \in E \Rightarrow (u = v \land r = s) \)
\( \triangleright \) For each input string \( x \), the output string \( y \) is obtained by concatenating \( \sigma(q) \) to the normal output for \( x \), where \( q \) is the last state reached through the analysis of \( x \)

PROPERTIES:

1. \( T_\tau \) is a function: \( X^* \to Y^* \)
2. Sequential \( \subset \) Subsequential Transduction \( \subset \) Finite State.
3. Input-output monotonicity (sequentiality) needs not be as strict as in STs.
Subsequential transducers: intuitive view

- **A SST relies on “delaying” the production of output symbols** until enough of the input sentence has been seen to guarantee a correct output

**An example of SST:**

```
un / a triangle and

y / triangle and

triangulo / λ

triangular / λ

square

grande / large triangle

λ

cuadrado / λ

grande / large square

... . . .
```

**Some translations:**

(un cuadrado y un triángulo grande, a square and a large triangle),
(un triángulo y un cuadrado, a triangle and a square),
(un triángulo grande, a large triangle),
(un cuadrado, a square),

... . . .

---

**Learning subsequential transducers:**

Onward subsequential transducer inference algorithm (OSTIA) [*Oncina, 91-93*]

1. Build an **“onward” tree representation** of the training data – called “OTST” (a tree in which output strings are as close to the root as possible)

2. Orderly traverse the tree, while **merging states** in order to get, hopefully, adequate generalizations

   - The traversal of the tree typically follows a **level by level order**

   - Two kinds of state merging:

     - **Local condition based**: involve only the two states under consideration; e.g.: *If both candidate states have the same output, or at least one has no output, merging is allowed*

     - **Derived merges**: once two states are merged, others may also need to be recursively merged in order to **preserve determinism**

   - The process may require to **“push-back” certain output substrings**

   - If a cascade of derived merges fails preserving determinism, the original and all the derived **merges are discarded**
An example of OSTIA state-merging process

\[ X = \{a, b\} \; ; \; Y = \{A, B\} \; ; \; T = \{(b, B), (a, AB), (bb, BA), (ba, BB), (aa, AAB)\} \]

![Diagram of TST(T) and OTST(T)]
An example of OSTIA state-merging process

\[ X = \{a, b\} \quad Y = \{A, B\} \quad T = \{(b,B), (a,AB), (bb,BA), (ba,BB), (aa,AAB)\} \]
Outline of the OSTIA [Oncina,91]

Algorithm OSTIA ("onward subsequential transducer inference algorithm")

Input: Finite set of (non ambiguous) input output pairs \( T \subset (X^* \times Y^*) \)

Output: Onward subsequential transducer \( \tau \) compatible with \( T \)

\[
\tau' = OTST(T); \quad \text{(let } Q(\tau') \text{ denote the set of states of } \tau')
\]

\[
\forall q \in Q(\tau') - \{q_0\} \text{ in a level-by-level order, do}
\]

\[
\forall p < q \text{ do}
\]

\[
\tau = \text{merge}(\tau', p, q)
\]

\[
\text{while } \exists q', q'' \in Q(\tau) \text{ that violate subsequential conditions, do}
\]

\[
\text{– try to restore subsequentiality by derived merging,}
\]

\[
\text{possibly requiring to "push-back" some output substrings of the edges incoming to } q', q'' \text{ towards the leaves of } \tau
\]

\[
\text{– if "derived merging" possible then } \tau = \text{merge}(\tau, q', q'')
\]

\[
\text{end while}
\]

\[
\text{if subsequential}(\tau) \text{ then } \tau' = \tau
\]

\[
\text{end } \forall p
\]

\[
\text{end } \forall q
\]

\[
\text{end OSTIA}
\]

Properties of OSTIA learning

[Oncina, García & Vidal, 93]

- **Correctness**: the resulting transducer is subsequential and is a (state-merging) generalization of the set of training pairs \( T \)

- **Convergence**: Using OSTIA the class of total Subsequential Transductions can be identified in the limit

- **Efficiency**: OSTIA average running time is observed to be \( O(n(m + k)) \), where

\[
- n = \sum_{(x,y) \in T} |x|, \text{ (overall length of input strings)}
\]

\[
- m = \max_{(x,y) \in T} |x| \text{ (longest output string)}
\]

\[
- k = |X| \text{ (size of input alphabet)}
\]

\[\Rightarrow \text{ huge sets of training examples can be easily handled}\]
Applications of SSTs and OSTIA learning

- Learning several toy but not trivial transduction tasks [Oncina, 91-93]
  - Simple arithmetic (e.g., decimal division by a fixed number)
  - Conversion of (large) English numbers into decimal notation
  - Translation of (large) English numbers into Spanish (and vice versa)
  - Conversion of Roman numbers into decimal
  - etc.

- Semantic decoding:
  - MLA [Castellanos et al.,98]
    (Translation of (pseudo-)NL sentences into predicate logic formulae)
  - (Subset of) ATIS [Vidal,94]
    (Translation of NL sentences into a DB access language)

- Language translation:
  - To be discussed here . . .

A simple experimental MT task: MTA

- Based on MLA (description and manipulation of simple visual scenes), winch was originally introduced as a challenging Language Learning task with a fairly simple syntax and small lexicon (about 30 words) [Feldman et al., 90]

- Reformulated for Machine Translation and extended, as required, to study the impact of increasing degree of input-output non-monotonicity, vocabulary size, etc. [Castellanos et al., 94]

Examples (Spanish-English):

un cuadrado mediano y claro y un círculo tocan a un círculo claro y un cuadrado mediano
a medium light square and a circle touch a light circle and a medium square

se añade un triángulo grande y oscuro muy a la izquierda del cuadrado y del círculo
a large dark triangle is added far to the left of the square and the circle

se elimina el círculo grande que esta encima del cuadrado y del triángulo mediano
the large circle which is above the square and the medium triangle is removed
MTA translation results using OSTIA

[Castellanos, Galiano and Vidal, ICGI–94], [Oncina et al., ICSNLP–94]

Spanish-English translation word error rates (TWER) for the extended MTA task, as a function of the training set size supplied to OSTIA. Test Set: 10,000 independent text input sentences.

<table>
<thead>
<tr>
<th>Train. Size</th>
<th>TWER(%)</th>
<th>States</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>58.8</td>
<td>412</td>
<td>1652</td>
</tr>
<tr>
<td>2,000</td>
<td>57.0</td>
<td>846</td>
<td>3197</td>
</tr>
<tr>
<td>4,000</td>
<td>51.8</td>
<td>1598</td>
<td>5970</td>
</tr>
<tr>
<td>8,000</td>
<td>3.4</td>
<td>186</td>
<td>891</td>
</tr>
<tr>
<td>16,000</td>
<td>0.0</td>
<td>17</td>
<td>206</td>
</tr>
</tbody>
</table>

- Convergence starts from 4,000–8,000 training pairs (decreasing size of the learned transducers)
- Good results achieved with very compact transducers learned from reasonably small training sets

▷ Bad news: These SSTs perform very poorly with imperfect text or speech input

“Good” basic SSTs can accept incorrect input producing even more incorrect output!

OSTIA learning generalizes the training pairs as much as possible, while preserving the input-output mapping represented by these pairs. This may lead to compact and accurate transducers but they generally involve excessive over-generalization of the input and output sentences.

- debajo izquierda esta por → square is removed
- elimina un y → the a
- a y y claro que → light square triangle which is
- muy esta oscuro → dark square which is square

Examples of Spanish sentences accepted (and translated) by a “good” transducer learned by OSTIA (0.0% translation WER for clean text input).

This is not a problem for translating clean text but it leads to very large translation errors for corrupted text or for speech input!
Helping OSTIA with input/output syntactic constraints

Two kind of conditions for OSTIA state merging:

- **Local conditions**: involve only the two states under consideration.
  
  Basic OSTIA allows merging two candidate states if both have the same output or at least one has no output [Oncina, 91-93].

- **Derived merges**: once two states have been merged, others may also need to be merged (while possibly “pushing-back” some output substrings) in order to preserve determinism.

**New Local Conditions:**

Use finite state models of the input (or domain) and/or the output (or range) to enforce input and/or output syntactic constraints

Idea [Oncina, 93-94]: **disallow the merging of two states if they correspond to different states of the input or output models**

The resulting algorithm is known as OSTIA-DR [Oncina, 93]
**Using input/output syntactic constraints:**

**Outline of OSTIA-DR [Oncina et al.,94]**

**Algorithm** OSTIA-DR ("OSTIA assisted by DOMAIN/RANGE constraints")

**Input:** Finite set of (non ambiguous) input output pairs \( T \subset (X^* \times Y^*) \)

finite state models, \( G_D, G_R \), of the domain \( X^* \) and range \( Y^* \)

**Output:** Onward subsequential transducer \( \tau' \) compatible with \( T \)

**Method:**

\[
\tau' = OTST(T); \quad (\text{let } Q(\tau') \text{ denote the set of states of } \tau')
\]

\[
\forall q \in Q(\tau') - \{q_0\} \text{ in a level-by-level order, do}
\]

\[
\forall p < q \quad \text{if } p, q \text{ are compatible with } G_D \text{ and/or } G_R \text{ do}
\]

\[
\tau = \text{merge}(\tau', p, q)
\]

**while** \( \exists q', q'' \in Q(\tau) \) that violate subsequential conditions, **do**

- try to restore subsequentiality by derived merging,
  - possibly requiring to "push-back" some output substrings of the edges incoming to \( q', q'' \) towards the leaves of \( \tau' \)
- if "derived merging" possible **then** \( \tau = \text{merge}(\tau, q', q'') \)

**end while**

**if subsequential(\( \tau \)) then** \( \tau' = \tau \)

**end \( \forall p \)**

**end \( \forall q \)**

**end OSTIA**

---

**Spanish-English MTA: OSTIA and OSTIA-DR learning results**

Translation word error rates (TWER) for the extended MTA Feldman’s task, as a function of the training set size supplied to OSTIA and OSTIA-DR (with 4-Gram input and output language models)

Test Set: 10,000 independent input sentences.

<table>
<thead>
<tr>
<th>Training set size</th>
<th>OSTIA</th>
<th>OSTIA-DR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TWER(%)</td>
<td>States</td>
</tr>
<tr>
<td>1,000</td>
<td>58.8</td>
<td>412</td>
</tr>
<tr>
<td>2,000</td>
<td>57.0</td>
<td>846</td>
</tr>
<tr>
<td>4,000</td>
<td>51.8</td>
<td>1598</td>
</tr>
<tr>
<td>8,000</td>
<td>3.4</td>
<td>186</td>
</tr>
<tr>
<td>16,000</td>
<td>0.0</td>
<td>17</td>
</tr>
</tbody>
</table>

Using Input/Output syntactic constraints, translation errors can be reduced by a factor of two.
Basic OSTIA–learned SST for Spanish-English MTA

OSTIA-DR–learned SST for Spanish-English MTA

(using both Domain and Range 3-Gram constraints)
MTA OSTIA and OSTIA-DR learning: impact of noisy text input and input–output language syntactic constraints

Spanish-English translation word error rates (TWER in %) of distorted test sentences for the extended MTA task, as a function of the training set size supplied to OSTIA and OSTIA-DR (with 4-Gram input and output language models). Noisy input translations obtained using error-correcting parsing.

<table>
<thead>
<tr>
<th>Train.Set Size</th>
<th>OSTIA Clean</th>
<th>OSTIA 5%Dist</th>
<th>OSTIA-DR Clean</th>
<th>OSTIA-DR 5%Dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>8,000</td>
<td>3.4</td>
<td>15.0</td>
<td>1.4</td>
<td>2.7</td>
</tr>
<tr>
<td>16,000</td>
<td>0.0</td>
<td>11.7</td>
<td>0.0</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Using Input/Output syntactic constraints increases robustness dramatically

Index

1 Probabilistic statement of MT problems ▷ 3
2 Rational or finite-state transduction (FST) ▷ 9
3 Subsequential Transduction: “OSTI” algorithm ▷ 15
4 Using input/output syntactic constraints: OSTIA-DR ▷ 30
5 OSTIA-DR: improving scalability ▷ 37
6 Statistical alignments: hybrid methods ▷ 45
7 Alignment-controlled state merging: OMEGA ▷ 48
8 Alignments and bilingual segmentation: GIATI ▷ 52
9 Speech-to-speech and computer assisted translation ▷ 61
10 GIATI revisited: pure statistical learning ▷ 66
Scalability issues

Subsequential transduction copes with input-output non-monotonicity by *delaying the decision for output (sub)strings*.

A training pair and a corresponding SST:

\( (\textit{se elimina un triángulo grande y claro}, \text{ a large light triangle is removed}) \)

\[ \begin{align*}
\text{se} &/\lambda \text{ elimina} &/\lambda \text{ un} &/\lambda \text{ a} &/\lambda \text{ triangulo} &/\lambda \text{ grande} &/\lambda \text{ y} &/\lambda \text{ claro} &/\lambda
\end{align*} \]

**Problem:**

The number of states can grow as much as \( O(n^k) \), where \( n \) is the number of functionally equivalent input words and \( k \) is the number of word–positions to be delayed.

The required amount of training data can become prohibitive.

Dealing with increasing vocabulary size \((n)\) and degree of non-monotonicity \((k)\)

**Approaches:**

\( k \Rightarrow \textbf{Partial alignment and word reordering} \)

[Vilar, Vidal, Amengual, Llorens, ECAI-96, SPECOM-96]:

Only preliminary studies carried out so far. Results suggest that *training-data requirements can be reduced dramatically.*

Further work required – related to *hybrid methods.*

\( n \Rightarrow \textbf{Bilingual categorization} \)

[Vilar, Marzal, Vidal, 95] [Amengual et al. Machine Translation, 2000]:

*While the direct approach degrades rapidly with increasing vocabulary sizes, categorization largely prevents accuracy degradation.*
Cutting down the impact of increasing vocabulary size through bilingual categorization

- Substitute words or groups of words by labels representing their syntactic (or semantic) categories within a limited rank of options.

- Learn a transducer with the categorized sentences, which entails a (much) smaller effective vocabulary.

- Expand each category-labeled edge of the learned transducer with a (small) transducer for this category.

Expansion leads to a single, perhaps large transducer which encompasses all the required information.

Categorization helps achieving adequate generalizations and proves very effective to prevent degradation of results with increasing vocabulary sizes.

A more complex and practical application: the “traveler task”

- Domain: human-to-human communication situations in the front-desk of a hotel

- Data produced semi-automatically on the base of a small “seed corpus” obtained from several traveler-oriented booklets

- Three language pairs: Spanish-English, Spanish-German and Spanish-Italian (only Spanish-English results reported here; similar results for the other languages)
The traveler task: features and examples

[Vidal et al., 96] (EuTrans ESPRIT project – first-phase)

Different sentence pairs in the corpus 171,481
Input/output vocabulary sizes 689 / 514
Average input/output sentence lengths 9.5 / 9.8
Input/output (2-Gram) test-set perplexities 6.8 / 5.6

(Similar features for Spanish-German and Spanish-Italian corpora)

Examples (Spanish-English):

Reservé una habitación individual y tranquila con televisión hasta pasado mañana.
I booked a quiet, single room with a tv. until the day after tomorrow.

Despiértanos mañana a las ocho menos cuarto, por favor.
Wake us up tomorrow at a quarter to seven, please.

Por favor, prepárenos nuestra cuenta de la habitación dos veintidós.
Could you prepare our bill for room number two two two for us, please?

Traveler task text-input experiments

[Vidal et al., 96] (EuTrans – first-phase final report)

OSTIA–DR learning using input and output 3-Gram LM constraints, with and without categorization into 7 categories: times-of-day, dates, room-numbers, etc.

Test-Set: 2,730 different sentences.

▷ Categorization leads to useful accuracy using moderate amounts of training data.
Traveler task error-correcting experiments

- OSTIA–DR learning using input/output 3–Gram LMs,
- Error model parameters estimated from artificially distorted input sentences, through EM and Viterbi re-estimation.

Training-data demands can be reduced by a factor of 2-3.

Index

1 Probabilistic statement of MT problems ▷ 3
2 Rational or finite-state transduction (FST) ▷ 9
3 Subsequential Transduction: “OSTI” algorithm ▷ 15
4 Using input/output syntactic constraints: OSTIA-DR ▷ 30
5 OSTIA-DR: improving scalability ▷ 37
6 Statistical alignments: hybrid methods ▷ 45
7 Alignment-controlled state merging: OMEGA ▷ 48
8 Alignments and bilingual segmentation: GIATI ▷ 52
9 Speech-to-speech and computer assisted translation ▷ 61
10 GIATI revisited: pure statistical learning ▷ 66
Statistical alignments and finite-state models

- Finite state transducer learning techniques seem to require large amounts of training data to produce acceptable results.

- Some byproducts of statistical alignment model training can be useful to improve the learning capabilities of finite state methods:
  - **Sentence-to-sentence word alignments**
  - **Word-to-word mappings (statistical dictionaries)**

[Brown et al. Computational Linguistics, 1990]: Decomposing \( \Pr(x \mid y) \) using bilingual word-position mappings or “alignments” as hidden variables:

\[
\Pr(x \mid y) = \sum_{a \in \mathcal{A}(y,x)} \Pr(x, a \mid y)
\]

where, \( \Pr(x, a \mid y) \) is mainly modeled by means of position alignment probabilities, e.g.: \( \Pr(i \mid j, I, J) \), and a statistical dictionary: \( \Pr(x_j \mid y_i) \)

Statistical alignment models

- **Alignments**: \( a \subseteq \{1, \ldots, I\} \times \{1, \ldots, J\}, \quad I = |x|, \quad J = |y| \)
- **Restriction**: \( a : \{1, \ldots, J\} \rightarrow \{0, \ldots, I\} \),

where \( a_j = 0 \) states that the \( j \)-th. position in \( y \) is not aligned with any position in \( x \)

**Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>per favore vorrei una camera doppia</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I (0) would (3) like (3) a (4) double (6) room (5) please (2)

\( a_1 = 0 \quad a_2 = 3 \quad a_3 = 3 \quad a_4 = 4 \quad a_5 = 6 \quad a_6 = 5 \quad a_7 = 2 \)

per favore vorrei una camera doppia

I would like a double room please
Adding local conditions for OSTIA-DR state merging: OMEGA

• OSTIA: only considers the output of the states: if both outputs are the same or at least one has no output, the join is possible [Oncina, 91-93].

• OSTIA-DR: also takes into account two Language Models (LM), one for the input (or domain) and one for the output (or range): two states cannot be joined if they correspond to different states of the input or output LMs [Oncina, 94-96].

• OMEGA [Vilar, ICGI-2000]: further takes into account alignments and word to word dictionaries.
The OMEGA extension to OSTIA

[Vilar, ICGI-2000]

- The initial tree is built taking alignments and/or dictionaries into account to avoid premature output. Each state $p$ is labeled with two sets:
  - $G(p)$ representing those words which are “guaranteed”, i.e., they will appear in the output of any path passing through $p$
  - $N(p)$ representing those words that “need” to be seen, i.e., those which have not appeared so far, but which should appear in the translation of at least one of the paths departing from $p$

- Local compatibility rules of OSTIA-DR now further include avoiding the join of two states $p$ and $q$ if $N(p) \cup N(q) \not\subseteq G(p) \cap G(q)$

- $N$ and $G$ can be derived from (probabilistic) dictionaries and/or alignments

- Input-Output Syntactic Constraints can be applied as in the original version of OSTIA(-DR)

OMEGA learning results

Spanish-English experiments; similar for Spanish-German [Vilar,98]

- **Data:** A subset of Spanish-English EuTrans-I traveler task data
  - Created by selecting those sentences with at most ten words
  - Test-Set: 588 different sentences, disjoint with training data

- **Training:** OMEGA versus OSTIA-DR
  - Bigram input and output syntactic constraints. No Categorization
  - Alignments obtained using the MAR statistical model

- **Search:** Error correcting parsing

<table>
<thead>
<tr>
<th>Different training pairs</th>
<th>OSTIA-DR TWER(%)</th>
<th>OMEGA-DR TWER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>27,28</td>
<td>16,51</td>
</tr>
<tr>
<td>2,000</td>
<td>19,64</td>
<td>11,17</td>
</tr>
<tr>
<td>4,000</td>
<td>11,88</td>
<td>8,33</td>
</tr>
<tr>
<td>8,000</td>
<td>8,31</td>
<td>5,57</td>
</tr>
<tr>
<td>16,000</td>
<td>5,19</td>
<td>4,16</td>
</tr>
</tbody>
</table>

▶ Training data demands can be reduced by a factor of 2
▶ Results improve using bilingual categorization
Index

1 Probabilistic statement of MT problems ▷ 3
2 Rational or finite-state transduction (FST) ▷ 9
3 Subsequential Transduction: “OSTI” algorithm ▷ 15
4 Using input/output syntactic constraints: OSTIA-DR ▷ 30
5 OSTIA-DR: improving scalability ▷ 37
6 Statistical alignments: hybrid methods ▷ 45
7 Alignment-controlled state merging: OMEGA ▷ 48
8 Alignments and bilingual segmentation: GIATI ▷ 52
9 Speech-to-speech and computer assisted translation ▷ 61
10 GIATI revisited: pure statistical learning ▷ 66

Regular Grammars and finite state transducers: a morphism theorem

**Theorem [Berstel 1979]:**

\[ T \subseteq X^* \times Y^* \text{ is a rational translation if and only if there exist an alphabet } Z, \text{ a regular language } L \subseteq Z^* \text{ and two morphisms } h_X : Z^* \rightarrow X^* \text{ and } h_Y : Z^* \rightarrow Y^* \text{ such that } T = \{(h_X(w), h_Y(w)) \mid w \in L\} \]

This theorem has suggested the development of a number of transducer learning techniques, including GIATI [Casacuberta, ICGI-2000]
Explicit use of statistical alignments for FST learning: GIATI

General idea in three steps:

1. Use sentence-to-sentence word alignments to convert each training pair \((x, y)\) of input/output sentences from \(X^* \times Y^*\) into a single training string \(z\) over an alphabet of “extended symbols” \(Z\) (composed of pairs of input/output symbols/strings).

2. Use an adequate grammar learning technique (e.g., N-Grams) to obtain a finite state “language model” for these strings.

3. Using the adequate morphisms, convert back each extended symbol of this model into a pair of input/output symbols/strings. This effectively transforms the language model into a finite state transducer.

This general method is referred to as Grammatical Inference and Alignments for Transducer Inference (GIATI).

GIATI: First step (Example)

Using statistical alignments to convert training pairs into training strings

Training pairs \((x, y) \in X^* \times Y^*\):
- una camera doppia → a double room
- una camera → a room
- la camera singola → the single room
- la camera → the room

Aligned sentences:
- una camera doppia
- a (1) double (3) room (2)
- una camera
- a (1) room (2)
- la camera singola
- the (1) single (3) room (2)
- la camera
- the (1) room (2)

Training strings \(z \in Z^*\):
- una+a camera doppia+double+room
- una+a camera+room
- la+the camera singola+single+room
- la+the camera+room
**GIATI: Second step**

*From training strings to grammars: n-grams*

\[
\Pr(z) \approx \prod_{i=1}^{\mid z \mid} \Pr(z_i | z_{i-n+1}, \ldots, z_{i-1})
\]

**PROBLEM:** Events unseen in the training set

**COMMON SOLUTION:** Smoothing techniques

**GIATI: Third step**

*From grammars to transducers: inverse labeling morphisms*

<table>
<thead>
<tr>
<th>GRAMMAR</th>
<th>TRANSDECER</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (q, a + b_1 + b_2 + \ldots + b_k, q') )</td>
<td>( (q, a, b_1 b_2 \ldots b_k, q') )</td>
</tr>
</tbody>
</table>

October 2004

E. Vidal, F. Casacuberta – ITI-UPV-DSIC
Comparative experiments: benchmark corpora

**EuTRANS-I corpus [Vidal 1997]**

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>10,000</td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>97,131</td>
<td>99,292</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>686</td>
<td>513</td>
</tr>
<tr>
<td>Test:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>2,996</td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>35,023</td>
<td>35,590</td>
</tr>
<tr>
<td>Bigram Perplexity</td>
<td>8.6</td>
<td>5.2</td>
</tr>
</tbody>
</table>

Semiautomatically generated Spanish-English sentences, human-to-human communication at a reception desk of a hotel

**EuTRANS-II corpus [ITI 2000]**

<table>
<thead>
<tr>
<th></th>
<th>Italian</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>3,038</td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>55,302</td>
<td>64,176</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>2,459</td>
<td>1,712</td>
</tr>
<tr>
<td>Test:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentences</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>6,121</td>
<td>7,243</td>
</tr>
<tr>
<td>Bigram Perplexity</td>
<td>31</td>
<td>25</td>
</tr>
</tbody>
</table>

Transcriptions of Italian-English spontaneous sentences, person-to-person communication in the hotel framework

**OSTIA / OMEGA / GIATI comparative results**

[EuTrans Final Report, 2000], [EuTrans D2.1a, 2000], [Casacuberta, 2002]

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Method</th>
<th>Assisted by</th>
<th>n-grams</th>
<th>TWER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EuTRANS-I</td>
<td>OSTIA</td>
<td>ECP</td>
<td>2</td>
<td>8.3</td>
</tr>
<tr>
<td>EuTRANS-I</td>
<td>OMEGA</td>
<td>ECP, IBM2'</td>
<td>2</td>
<td>6.6</td>
</tr>
<tr>
<td>EuTRANS-I</td>
<td>OMEGA</td>
<td>ECP, IBM2, ABC</td>
<td>2</td>
<td>3.9</td>
</tr>
<tr>
<td>EuTRANS-I</td>
<td>GIATI</td>
<td>BOS, IBM5</td>
<td>5</td>
<td>6.6</td>
</tr>
<tr>
<td>EuTRANS-II</td>
<td>OMEGA</td>
<td>ECP, IBM2</td>
<td>2</td>
<td>41.7</td>
</tr>
<tr>
<td>EuTRANS-II</td>
<td>OMEGA</td>
<td>ECP, IBM2, ABS</td>
<td>2</td>
<td>36.5</td>
</tr>
<tr>
<td>EuTRANS-II</td>
<td>GIATI</td>
<td>BOS, IBM5</td>
<td>2</td>
<td>28.1</td>
</tr>
<tr>
<td>EuTRANS-II</td>
<td>GIATI</td>
<td>BOS, IBM5, ABS</td>
<td>2</td>
<td>24.9</td>
</tr>
</tbody>
</table>

ECP = Error-Correcting Parsing  
BOS = Back-Off Smoothing  
ABS = Automatic Bilingual Segmentation  
ABC = Automatic Bilingual Categorization  
IBMk = IBM Model k statistical alignments  
IBM2’ = Symmetrized IBM2
## Summary of results

Translation Word Error Rate (TWER %)

<table>
<thead>
<tr>
<th>Task</th>
<th>MLA</th>
<th>EuTRANS-0</th>
<th>EuTRANS-I</th>
<th>EuTRANS-II</th>
<th>TT2-XRCE</th>
<th>AMETRA</th>
<th>TT2-UE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sp-En</td>
<td>Sp-En</td>
<td>Sp-En</td>
<td>It-En</td>
<td>En-Sp</td>
<td>Sp-Ba</td>
<td>En-Sp</td>
</tr>
<tr>
<td>Languages</td>
<td>30</td>
<td>689/514</td>
<td>689/514</td>
<td>2.5K/1.7K</td>
<td>26K/30K</td>
<td>719/1.3K</td>
<td>84K/97K</td>
</tr>
<tr>
<td>Vocabularies</td>
<td>110K</td>
<td>4.5M</td>
<td>100K</td>
<td>50K</td>
<td>600K</td>
<td>90K</td>
<td>6M</td>
</tr>
<tr>
<td>Year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSTIA</td>
<td>3</td>
<td>≈1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OSTIA-DR</td>
<td>1</td>
<td>&lt;1</td>
<td>10</td>
<td>&gt;80</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OMEGA</td>
<td>-</td>
<td>&lt;1</td>
<td>4</td>
<td>37</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GIATI</td>
<td>-</td>
<td>3</td>
<td>7</td>
<td>25</td>
<td>32</td>
<td>40</td>
<td>56</td>
</tr>
<tr>
<td>Best result</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>25</td>
<td>28</td>
<td>36</td>
<td>47</td>
</tr>
<tr>
<td>Non FS system</td>
<td>-</td>
<td>-</td>
<td>AT</td>
<td>AT</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

Languages: **English**, **Spanish**, **Italian**, **Basque**

PB = Phrase-based alignment models

AT = Alignment Templates

---

**Index**

1. Probabilistic statement of MT problems ▶️ 3
2. Rational or finite-state transduction (FST) ▶️ 9
4. Using input/output syntactic constraints: OSTIA-DR ▶️ 30
5. OSTIA-DR: improving scalability ▶️ 37
6. Statistical alignments: hybrid methods ▶️ 45
7. Alignment-controlled state merging: OMEGA ▶️ 48
8. Alignments and bilingual segmentation: GIATI ▶️ 52
9. *Speech-to-speech and computer assisted translation* ▶️ 61
10. GIATI revisited: pure statistical learning ▶️ 66
Speech input machine translation using SFSTs

Given an utterance $v$ from the source language, search for a target sentence:

$$\hat{y} = \arg\max_y P_r(y \mid v) \approx \arg\max_y \max_x (P_r(x, y) \cdot P_r(v \mid x))$$

- **Models:**
  - $P_r(v \mid x)$: Acoustic models HMMs
  - $P_r(x, y)$: Translation models SFST
- **Architectures:**
  - Integrated: Speech decoding and translation are performed simultaneously
  - Serial: First speech decoding and then translation of decoded speech
- **Search:** Viterbi algorithm
- **Assessment:**
  - **Word Error Rate:** Minimum number of edit operations to convert the decoded sentence into a reference source sentence
  - **Translation Word Error Rate:** Minimum number of edit operations to convert translated sentence into a reference target sentence.


---

Speech-input translation results

- For EuTRANS-0 and EuTRANS-I: Acoustic models of 26 Spanish monophone units were left-to-right CDHMMs (HTK Toolkit). 3.8h of training speech by 20 speakers (microphone and telephone speech data)
- For EuTRANS-II: Decision-tree clustered generalized triphones (CART with 1,500 tied states plus silence). 7.9h of training speech, uttered by 276 speakers

Speech-input **word error rate** and **translation word error rate** (WER/TWER %)

<table>
<thead>
<tr>
<th>Task acoustics</th>
<th>EuTRANS-0 microphone Sp-En</th>
<th>EuTRANS-0 telephone Sp-En</th>
<th>EuTRANS-I microphone Sp-En</th>
<th>EuTRANS-I telephone Sp-En</th>
<th>EuTRANS-II telephone It-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMEGA</td>
<td>4.1 / 3.5</td>
<td>8.4 / 7.6</td>
<td>13.6 / 12.5</td>
<td>11.6 / 17.7</td>
<td>22.1 / 49.4</td>
</tr>
<tr>
<td>GIATI</td>
<td>2.3 / 6.0</td>
<td>7.5 / 10.7</td>
<td>4.1 / 7.8</td>
<td>10.5 / 12.6</td>
<td>22.1 / 37.9</td>
</tr>
<tr>
<td>Best result</td>
<td>-</td>
<td>-</td>
<td>4.1 / 6.9</td>
<td>11.6 / 13.3</td>
<td>22.1 / 37.8</td>
</tr>
<tr>
<td>System</td>
<td>-</td>
<td>-</td>
<td>AT</td>
<td>AT</td>
<td>AT</td>
</tr>
</tbody>
</table>

AT = Alignment Templates

More results for Catalan-English, Spanish-Basque and Portuguese-English
Computer-assisted translation using SFSTs

Given a source sentence $x$ and a prefix of the target sentence $y_p$, search for a suffix of $y_p$:

$$
\arg\max_{y_s} \Pr(y_s | x, y_p) = \arg\max_{y_s} \Pr(x, y_p y_s)
$$

- **Model**: $\Pr(x, y_p y_s) \rightarrow$ SFST

- **Search**: Viterbi modified to take the given prefix $y_p$ into account

- **Assessment**: Key-Stoke Ratio (KSR): Number of key-strokes needed to produce a reference translation with the help of the prediction engine, divided by the number of running characters in the reference translation

Computer-assisted translation: Results

Key Stroke Ratio (KSR %)

<table>
<thead>
<tr>
<th>Task</th>
<th>XRCE</th>
<th>EU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>En-Sp</td>
<td>Sp-En</td>
</tr>
<tr>
<td>GIATI</td>
<td>15.6</td>
<td>18.9</td>
</tr>
<tr>
<td>AT</td>
<td>20.9</td>
<td>21.4</td>
</tr>
<tr>
<td>PB</td>
<td>11.9</td>
<td>13.7</td>
</tr>
</tbody>
</table>

PB = Phrase Based Models  AT = Alignment Templates

More results for German-English, French-English and Spanish-Basque
Index

1 Proabilistic statement of MT problems ▷ 3
2 Rational or finite-state transduction (FST) ▷ 9
3 Subsequential Transduction: “OSTI” algorithm ▷ 15
4 Using input/output syntactic constraints: OSTIA-DR ▷ 30
5 OSTIA-DR: improving scalability ▷ 37
6 Statistical alignments: hybrid methods ▷ 45
7 Alignment-controlled state merging: OMEGA ▷ 48
8 Alignments and bilingual segmentation: GIATI ▷ 52
9 Speech-to-speech and computer assisted translation ▷ 61
10 GIATI revisited: pure statistical learning ▷ 66

Pure statistical approach: GIATI revisited

Let $Pr(x, y)$ be the joint probability of a pair of sentences $(x, y)$

- Let $J$ and $I$ be the *given* lengths of $x$ and $y$, respectively.
- Assume that $y$ is segmented into $J$ segments,$\mu : \{1, ..., J\} \rightarrow \{1, ..., I\}$ with $\mu_{j+1} > \mu_j$ for $1 \leq j < J$ and $\mu_J = I$

Further assumptions:

- The distributions that rule $I$, $J$ and $\mu$ are uniform.
- The correspondence among source symbols and target segments is monotone.
- By using a $n$-grams approximation with an special “end” symbol $\$.$

$$Pr(x, y) \propto \sum_{K} \sum_{\mu_1^K} \prod_{k=1}^J Pr(x_k, y_{\mu_k-1+1}^{\mu_k-1} | x_{k-n+1}^{k-1}, y_{\mu_k-1+1}^{\mu_k-1}) \cdot Pr(\$, \$ | x_{J-n+2}^J, y_{\mu_J+n+2}^\mu_J)$$
Pure statistical approach: GIATI revisited

Features:

- Main feature: All possible segmentations of the training set are considered.
- Parameter estimation: E-M algorithm.
- A SFST implementation:
  - The states are all possible \((x_{k-n+1}^{k}, y_{\mu_{k-n+1}}^{\mu_{k}})\) in the training set;
  - The probability of a transition between two states \((x_{k-n+2}^{k}, y_{\mu_{k-n+2}}^{\mu_{k}})\) and \((x_{k-n+1}^{k-1}, y_{\mu_{k-n+1}}^{\mu_{k-1}})\) is \(P_r(x_k, y_{\mu_{k-1}}^{\mu_{k}} | x_{k-n+1}^{k-1}, y_{\mu_{k-n+1}}^{\mu_{k-1}})\) with \(x_k\) as source symbol and \(y_{\mu_{k-n+1}}^{\mu_{k-1}}\) as the target string;
  - The probability that \((x_{k-n+1}^{k-1}, y_{\mu_{k-n+1}}^{\mu_{k-1}})\) of a final state is \(P_r(\$, \$ | x_{k-n+1}^{k-1}, y_{\mu_{k-n+1}}^{\mu_{k-1}})\).
- Generalization to arbitrary segmentations of the source sentence.

Conclusions

- We have thoroughly explored the learning of FST and its applications in MT
- Other contributions in this area: [Knight & Al-Onaizan, 98], [Mäkinen, 99], [Bangalore, Ricardi et al., 01]
- As task complexity and/or data scarceness increases, it becomes more and more important to make use of methods borrowed from statistical language processing.
  Particularly relevant: statistical alignments and smoothing techniques
- Making explicit use of these techniques, GIATI is among the most promising approaches for FST MT
- A new pure statistically based development of GIATI is under way