Pattern Recognition approaches to Machine Translation

Francesc Casacuberta and Enrique Vidal

Pattern Recognition and Human Language Technology Group
Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain

January 2005

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

Index

1. Introduction to Machine Translation,
   Statistical Framework for Machine Translation (Mon. E.Vidal)
2. Statistical Alignment Models (Mon. F.Casacuberta)
3. Advanced Statistical Alignment Models (Tue. F.Casacuberta)
4. Stochastic Finite-State Translation Models (Tue. E.Vidal)
5. Phrase-based Alignment Models and Alignment Templates (Wed. F.Casacuberta)
7. Finite-State Translation Models based on Alignments (Thu. E.Vidal)
8. Recursive Alignment Models (Thu. F.Casacuberta)
9. Speech-to-Speech Translation (Fri. F.Casacuberta)
10. Computer-Assisted Translation (Fri. E.Vidal)
Seminars on Formal Syntax and Semantics  
Universitat Rovira i Virgili

Pattern Recognition approaches to Machine Translation

F. Casacuberta and E. Vidal

Pattern Recognition and Human Language Technology Group  
Instituto Tecnológico de Informática  
Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia, Spain

Introduction to Machine Translation

Enrique Vidal

evidal@iti.upv.es

January 2005

E. Vidal – ITI-UPV-DSIC

Index

1 Objectives of Machine Translation (MT) ▷ 2
2 Approaches to MT ▷ 7
3 Linguistic Resources ▷ 10
4 Assessment ▷ 12
5 Limited Domains ▷ 14
6 Speech-to-speech MT ▷ 19
7 Computer Assisted Translation ▷ 22
8 Brief History of MT ▷ 25

Bibliography:

D. Arnold, L. Balkan, R. Lee Humphreys, S. Meijer, L.Sadler:  
“Machine Translation, an introductory guide”. NCC Blackwell, 1994
MT objectives: Erroneous conceptions

- MT is a waste of time because a machine never will translate Shakespeare.

- Generally, the quality of translation you can get from an MT system is very low.

- MT threatens the jobs of translators

- There is an MT system that translates what you say into Japanese and translates the other speaker’s replies in English.

- There is an amazing South American Indian language with a structure of such logical perfection that it solves the problem of design MT systems.

- MT systems are machines, and buying an MT system should be very much like buying a car.
MT objectives: Facts

- MT is useful.
- There are many situations that MT systems produce reliable, if less than perfect, translations at high speed.
- In some circumstances, MT systems can produce good quality outputs.
- MT does not threaten translators’ jobs: High demand of translations and too repetitive translation jobs.
- Speech-to-speech MT is still a research topic.
- There are many open research problems in MT.
- Building a traditional MT system is a time consuming job.
- A user will typically have to invest a considerable amount of effort in customizing an MT system.

Need of pre/post-editing

- While the number of errors and bad constructions is high, “post-editing” can make the result useful.
- Many problems could have been avoided by making the source text “simpler”.
- Simplification of the translation problem by using adequate rules to produce “controled” (i.e., simple and regular) source text.
General scheme for MT

source

PRE-EDITING

TRANSLATOR

POST-EDITING

target

Index

1 Objectives of Machine Translation (MT) ▶ 2

2 Approaches to MT ▶ 7

3 Linguistic Resources ▶ 10

4 Assessment ▶ 12

5 Limited Domains ▶ 14

6 Speech-to-speech MT ▶ 19

7 Computer Assisted Translation ▶ 22

8 Brief History of MT ▶ 25
Approaches to MT: Analysis detail

- Depth of analysis
- Generation

INTERLINGUA
TRANSFER
DIRECT
TARGET

Approaches to MT: Technologies

- (Linguistic) knowledge-based methods
- (Memorized) example-based methods
  - Translation memories
- Statistical models
  - Alignment models
  - Finite-State models
- Hybrid models
Index

1 Objectives of Machine Translation (MT) ▷ 2
2 Approaches to MT ▷ 7
3 Linguistic Resources ▷ 10
4 Assessment ▷ 12
5 Limited Domains ▷ 14
6 Speech-to-speech MT ▷ 19
7 Computer Assisted Translation ▷ 22
8 Brief History of MT ▷ 25

Linguistic resources

• Dictionaries
• Grammars
• Corpora
• Paragraph-aligned and Labeled Corpora
Index

1 Objectives of Machine Translation (MT) 1
2 Approaches to MT 7
3 Linguistic Resources 10
4 Assessment 12
5 Limited Domains 14
6 Speech-to-speech MT 19
7 Computer Assisted Translation 22
8 Brief History of MT 25

Assessment

- Test sentences
- Subjective evaluation based on the number of words that need to be corrected or deleted
- Test sentences with reference translation
- Automatic assessment
  - Editing Distances:
    - Translation Word Error Rate (TWER)
  - Multireference TWER
  - N-Gram based: BLUE
Limited domains or “sublanguages”

- Tasks with small or medium-sized vocabularies and restricted semantic scope.
- Robust systems needed.
- Manual “post-editing” should be avoided or minimized.
- Only low development costs can be afforded.
Limited domain or “sublanguages”: Example

The “Traveler Task” [Vidal et al., 96] (EuTrans ESPRIT project – first-phase)

- Domain: human-to-human communication situations in the front-desk of a hotel.
- Three language pairs:
  - Spanish-English,
  - Spanish-German
  - Spanish-Italian

Features of the Spanish-English task (similar for the other language pairs)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input/output vocabulary sizes</td>
<td>∼ 700 / 500</td>
</tr>
<tr>
<td>Average input/output sentence lengths</td>
<td>∼ 10 / 10</td>
</tr>
<tr>
<td>Input/output test-set perplexities</td>
<td>∼ 11 / 6</td>
</tr>
</tbody>
</table>

The Traveler Task: examples of Spanish-English paired sentences

<table>
<thead>
<tr>
<th>Spanish</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reservé una habitación individual y tranquila con televisión hasta pasado mañana.</td>
<td>I booked a quiet, single room with a tv. until the day after tomorrow.</td>
</tr>
<tr>
<td>Por favor, prepárenos nuestra cuenta de la habitación dos veintidós.</td>
<td>Could you prepare our bill for room number two two two for us, please?</td>
</tr>
</tbody>
</table>
Language translation and language understanding

- Under the Limited-Domain (LD) framework both Language Understanding (LU) and Language Translation (LT) can be properly formulated in a uniform way.

- The ultimate goal of a LD LU system is to drive the actions associated to the meaning conveyed by the sentences issued by the users.

- Since actions are to be performed by machines, the understanding problem can then be simply formulated as translating the natural language sentences into formal sentences of an adequate (computer) command language in which the actions to be carried out can be specified.

- Thus, LU can be seen as a specific (simpler) case of LT in which the output language is formal rather than natural.

Index

1 Objectives of Machine Translation (MT)  ▷ 2
2 Approaches to MT  ▷ 7
3 Linguistic Resources  ▷ 10
4 Assessment  ▷ 12
5 Limited Domains  ▷ 14
6 Speech-to-speech MT  ▷ 19
7 Computer Assisted Translation  ▷ 22
8 Brief History of MT  ▷ 25
Approaches to speech-to-speech translation

- **Traditional** → Serially couple the following (existing) devices:
  1. Conventional continuous word recognition front-end.
  2. Text-to-text, general-purpose, knowledge-based MT system (adapted by experts to the task in hand).
  3. Text-to-speech output language synthesizer.

- **Integrated approach** → Consider language translation as a global input-output decoding problem:
  1. Develop an integrated device that directly accepts speech (or text) input sentences and outputs corresponding sentences in the target language.
  2. Implement input-output decoding as a global optimization search that takes into account all the information compiled into the integrated recognition/translation device.
  3. Chose a translation model that is trainable from input-output translation examples.

Speech translation:
Advantages of integration and automatic learning

- *Tight integration* leads to speech-input translation systems which are significantly more robust, as compared with other based on the more traditional, *loosely coupled* approach.

- *Trainability* leads to *better adaptation* to specific domains at much *lower development costs*. 
Computer Assisted Translation (CAT)

- Do not attempt fully automated MT
- Aim at high-quality results
- Let the human translator fully command the process
- Allow for tight human-machine cooperation
- Aim to increase human translator productivity
- Ergonomic issues and multimodality: keyboard, mouse, speech, ...
**Typical Computer Assisted Translation Scenario**

Text prediction based on both the source-language text to be translated and preceding text that has been validated by the user.

For each source sentence or paragraph to be translated:

1. The system provides its best (or N-best) translation suggestion

2. The user selects a correct part (typically a prefix) of this suggestion and starts amending the remaining part or entering new text by him/herself

3. After each user-entered word (or key-stroke), the system recomputes its best suggestion(s), thereby starting a new human-system interaction cycle.

**Index**

1. Objectives of Machine Translation (MT) ▶ 2
2. Approaches to MT ▶ 7
3. Linguistic Resources ▶ 10
4. Assessment ▶ 12
5. Limited Domains ▶ 14
6. Speech-to-speech MT ▶ 19
7. Computer Assisted Translation ▶ 22

8. *Brief History of MT* ▶ 25
Brief history of MT

- **1949** Weaver: Information-theory based approach

- **1957** Chomsky: Natural language is not governed by statistics

- **1960** ALPAC (Automatic Language Processing Advisory Committee) report: No useful MT results are foreseen

- **1960-nowadays**
  - SYSTRAN system: based on dictionaries
  - Several (linguistic) knowledge-based approaches

- **1985-95** “Empiricists” methods are introduced: corpus-based and statistical approaches (IBM, 1989)

- **1995-nowadays** “Empiricists” methods are thriving. Speech-to-speech MT in limited domains

Recent history of MT: “Empiricists” methods

- **1989-95** Statistical approach to MT by IBM Yorktown Heights researchers
  - Corpus: Hansards
  - Parallel English/French transcriptions of parliamentary discussions
  - DARPA competitive assessment (1994): Results comparable to those achieved by traditional approaches

- **1990-05** Development of statistical techniques and other empiricists methods
  - Progress of the statistical approach (by IBM and other groups)
  - Other “example-based”, empiricist techniques: Memory-Based, Finite-State, etc.
  - Statistics are applied to other MT-related fields: Lexicography, syntactic labeling of corpora, etc.
  - Progress in Grammars and Syntactic Analysis
  - Computer Assisted Translation
Pattern Recognition approaches to Machine Translation

F. Casacuberta and E. Vidal

Pattern Recognition and Human Language Technology Group
Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain

Statistical Framework to Machine Translation

Enrique Vidal
evidal@iti.upv.es

January 2005

E. Vidal – ITI-UPV-DSIC

Index

1 Notation and background ▷ 2

2 Text-input machine translation ▷ 4

3 Speech-input machine translation ▷ 10

4 Computer-assisted translation ▷ 13

5 Bibliography ▷ 18
Notation and Basic Concepts

- $x$ and $y$ will generally denote source and target texts, respectively.

- **Conditional and Unconditional Probabilities:**
  \[
  \Pr(X = x \mid Y = y) \equiv \Pr(x \mid y), \quad \Pr(X = x) \equiv \Pr(x)
  \]

- **Bayes’ Rule:**
  \[
  \Pr(x \mid y) \cdot \Pr(y) = \Pr(y \mid x) \cdot \Pr(x)
  \]

- **Joint Probability:**
  \[
  \Pr(x, y) = \Pr(x) \cdot \Pr(y \mid x)
  \]

- \[
  \Pr(x) = \sum_y \Pr(x, y)
  \]

- \[
  \max_x \Pr(x) = \Pr(\arg\max_x \Pr(x))
  \]

- \[
  \sum_x \Pr(x) \approx \max_x \Pr(x)
  \]
Index

1 Notation and background ▶ 2
  ⊙ 2 Text-input machine translation ▶ 4
3 Speech-input machine translation ▶ 10
4 Computer-assisted translation ▶ 13
5 Bibliography ▶ 18

General Framework

• Every sentence $y$ in a target language is considered as a possible translation of any other sentence $x$ in another source language.

• For each possible pair of sentences $y, x$, there is a probability $\Pr(y \mid x)$.

• $\Pr(y \mid x)$ should be low for pairs $(y, x)$ such as:
  (una habitación con vistas al mar, are all expenses included in the bill?)

• $\Pr(y \mid x)$ should be high for pairs such as:
  (¿hay alguna habitación tranquila libre?, is there a quiet room available?)
A direct approach

Search for a target sentence with maximum posterior probability:

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

A “direct model”

Decompose \( \Pr(y \mid x) \) using Bayes’ rule:

\[ \hat{y} = \arg\max_y \Pr(y \mid x) = \arg\max_y \Pr(x \mid y) \cdot \Pr(y) \]

A “distorted channel model”

Need: alignment and lexicon models

An inverse approach

Decompose \( \Pr(y \mid x) \) using Bayes’ rule:

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

A “distorted channel model”

Need: a target-language model + alignment and lexicon models
A finite-state approach

The direct probability can be decomposed in a different way:

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

A “joint” model

![Diagram of a joint model](image)

A stochastic finite-state transducer can model the joint distribution

Translation search

- Direct approach: alignment and lexicon models
  $$\hat{y} = \arg\max_y \Pr(y \mid x)$$

- Inverse approach: a target-language model + alignment and lexicon models
  $$\hat{y} = \arg\max_y \Pr(x \mid y) \cdot \Pr(y)$$

- Joint approach: stochastic finite-state transducer
  $$\hat{y} = \arg\max_y \Pr(x, y)$$
Speech-input translation

Given an input acoustic sequence \( v \), search for a target sentence with maximum posterior probability:

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

But this can be seen as a “two-step process”:

\[
v \rightarrow x \rightarrow y
\]

where the “hidden variable” \( x \) accounts for all possible input decodings of \( v \):

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

(with the assumption: \( \Pr(v \mid x, y) \) does not depend on the target sentence \( y \))
Speech-input translation

\[
\arg\max_y \Pr(y \mid v) \approx \arg\max_y \max_x (\Pr(x, y) \cdot \Pr(v \mid x))
\]
\[
= \arg\max_y \max_x (\Pr(y) \cdot \Pr(x \mid y) \cdot \Pr(v \mid x))
\]
\[
= \arg\max_y \max_x (\Pr(x) \cdot \Pr(y \mid x) \cdot \Pr(v \mid x))
\]

- \(\Pr(v \mid x) \approx \text{ACOUSTIC MODELS}\)
- \(\Pr(x, y) \approx \text{FINITE-STATE TRANSDUCERS}\)
- \(\Pr(x \mid y), \Pr(y \mid x) \approx \text{ALIGNMENT AND LEXICON MODELS}\)
- \(\Pr(y) \approx \text{TARGET LANGUAGE MODELS}\)
- \(\Pr(x) \approx \text{SOURCE LANGUAGE MODELS}\)

Index

1 Notation and background 2
2 Text-input machine translation 4
3 Speech-input machine translation 10
4 Computer-assisted translation 13
5 Bibliography 18
Text prediction for Computer-Assisted Translation (CAT)

Given a source text $x$ and a “correct” prefix $y_p$ of the target text, search for a suffix $\hat{y}_s$, that maximizes the posterior probability over all possible suffixes:

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

Taking into account that $\Pr(y_p \mid x)$ does not depend on $y_s$, we can write:

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

Main difference with text-input machine translation: search over the set of suffixes.

Target language dictation in CAT

A human translator dictates the translation of a source text, $x$, producing a target language acoustic sequence $v$.

Given $v$ and $x$, the system should search for a most likely decoding of $v$:

$$\hat{y} = \arg\max_{y} \Pr(y \mid x, v)$$

By the assumption that $\Pr(v \mid x, y)$ does not depend on $x$,

$$\hat{y} = \arg\max_{y} \Pr(v \mid y) \cdot \Pr(x \mid y) \cdot \Pr(y)$$

- $\Pr(v \mid y) \approx \text{(TARGET LANGUAGE) ACOUSTIC MODELS}$
- $\Pr(x \mid y) \approx \text{TRANSLATION MODEL}$
- $\Pr(y) \approx \text{TARGET LANGUAGE MODEL}$

Similar to plain speech decoding, where: $\hat{y} = \arg\max_y \Pr(v \mid y) \cdot \Pr(y)$
Further use of speech recognition in CAT

Let $x$ be the source text and $y_p$ a “correct” prefix of the target sentence. As in pure text CAT the system suggests an optimal suffix:

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

(1)

The user is now allowed to utter some words, $v$, generally aimed at amending parts of $\hat{y}_s$ and the system has then to obtain a most probable decoding of $v$:

$$\hat{d} = \arg\max_d \Pr(d \mid x, y_p, \hat{y}_s, v).$$

(2)

Finally, the user can enter additional amendment keystrokes $k$, to produce a new consolidated prefix, $y_p$, based on the previous $y_p$, $\hat{d}$, $k$ and parts of $\hat{y}_s$.

Further use of speech recognition in CAT (cont.)

From Eq. (2):

$$\hat{d} = \arg\max_d \Pr(d \mid x, y_p, \hat{y}_s) \cdot \Pr(v \mid x, y_p, \hat{y}_s, d)$$

and, by making the assumption that $\Pr(v \mid x, y_p, \hat{y}_s, d)$ only depends on $d$:

$$\hat{d} = \arg\max_d \Pr(d \mid x, y_p, \hat{y}_s) \cdot \Pr(v \mid d)$$

- $\Pr(v \mid d) \approx \text{TARGET LANGUAGE ACOUSTIC MODELS}$
- $\Pr(d \mid x, y_p, \hat{y}_s) \approx \text{TARGET LANGUAGE MODEL CONSTRAINED BY THE SOURCE SENTENCE, THE PREFIX AND THE SUFFIX}$
Index

1. Notation and background ▷ 2

2. Text-input machine translation ▷ 4

3. Speech-input machine translation ▷ 10

4. Computer-assisted translation ▷ 13

5. Bibliography ▷ 18

Bibliography


2: Statistical Alignment Models

Francisco Casacuberta Nolla

fcn@iti.upv.es

24-28 January 2005

Index

1 Statistical framework to machine translation ▶ 2
2 Alignments ▶ 11
3 Statistical alignment models ▶ 20
4 First-order alignment models ▶ 50
5 Categorization in statistical modeling ▶ 66
6 Bibliography ▶ 74
General framework

• Every sentence $y$ in one language is a translation of any sentence $x$ in another language.

• For each possible pair of sentences, $y$ and $x$, there is a probability $P_r(y | x)$.

• The probability of pairs of sentences as
  *quiero una habitación doble con vistas al mar* # are all expenses included in the bill ?
  should be low.

• The probability of pairs of sentences as
  ¿ hay alguna habitación tranquila libre ? # is there a quiet room available ?
  should be high.
General framework

Given a source sentence \( x \), search for the sentence \( \hat{y} \)

\[
\hat{y} = \arg\max_y P_r(y | x)
\]

Approaches

- A direct approach: *maximum entropy models*
- An inverse approach: *channel models*

An inverse approach

*Given a source sentence* \( x \), search for the sentence \( \hat{y} \)

\[
\hat{y} = \arg\max_y P_r(y | x) = \arg\max_y P_r(x | y) \cdot P_r(y)
\]

A channel model

\[
\text{Pr}(y) \xrightarrow{y} \text{Pr}(x | y) \xrightarrow{x}
\]

A target-language model + alignment and lexicon models
Pattern Recognition approaches to Machine Translation

Statistical Alignment Models

Translation search

\[
\hat{y} = \arg\max_y P_r(x | y) \cdot P_r(y)
\]

- Inverse approach:
  - A target-language model: \( P_r(y) \approx P_r(y) \)
  - Translation models (alignment and lexicon models): \( P_r(x | y) \approx P_r(x | y) \)
  - Search procedure: \( \hat{y} = \arg\max_y P_r(x | y) \cdot P_r(y) \)

An inverse approach

\[
\hat{y} = \arg\max_y P_r(x | y) \cdot P_r(y)
\]
An inverse approach: The target language model

\[ \argmax_y P_r(x \mid y) \cdot P_r(y) \]

Language models

**Word n-grams**

\[ P_r(y) = \prod_{i=1}^{\lvert y \rvert} P_r(y_i \mid y \ldots y_{i-1}) \approx P_r(y) = \prod_{i=1}^{\lvert y \rvert} p_n(y_i \mid y_{i-n+1} \ldots y_{i-1}) \]

**n-grams of categories**

\[ P_r(y) \approx P_r(y) = \prod_{i=1}^{\lvert y \rvert} p_n(C_i \mid C_{i-N+1} \ldots C_{i-1}) \cdot p(y_i \mid C_i) \]

**Regular or context-free grammars**

\[ P_r(y) \approx P_r(y) = \sum_{d(y)} p_G(d(y)) \approx \max_{d(y)} p_G(d(y)) \]
Learning language models

- Probabilistic estimation techniques.
- Grammatical inference techniques.
- SMOOTHING.
- Extensions: cache, triggers, categories, etc.
- Widely used toolkits for $n$-grams:
  - SRILM - The SRI Language Modeling Toolkit
    http://www.speech.sri.com/projects/srilm/
  - The CMU Statistical Language Modeling (SLM) Toolkit
    http://www.speech.cs.cmu.edu/SLM_info.html

Index

1 Statistical framework to machine translation ▶ 2

2 Alignments ▶ 11

3 Statistical alignment models ▶ 20

4 First-order alignment models ▶ 50

5 Categorization in statistical modeling ▶ 66

6 Bibliography ▶ 74
An inverse approach

\[
\text{argmax}_{y} \ P_r(x | y) \cdot P_r(y)
\]

Example of word alignments

I have made a reservation for a quiet room for Rosario Cabedo.
Example of word alignments

H. Ney, *Statistical Natural Language Processing*, 2003: Canadian Hansards

Could you ask for a taxi, please?
Example of word alignments

AMETRA corpus

1996 de marzo de 20 a Lemoa En Lemoan

METEO corpus

sud meitat seva la en Llevant de des sobretot sobre todo desde Levante en su mitad sur
Alignments

- **Alignments**: (Brown et al. 90) \( J = |x| \ y = |y| \)

  \[ a \subseteq \{1, \ldots, J\} \times \{1, \ldots, I\} \]

  - Number of connections: \( I \cdot J \)
  - Number of alignments: \( 2^{I \cdot J} \)

- **Constrain**: \( a : \{1, \ldots, J\} \rightarrow \{0, \ldots, I\} \), \( (a_j = 0 \Rightarrow j \text{ in } x \text{ is not aligned with any position in } y) \).

  - Number of alignments: \( (I + 1)^J \)

- Set of possible alignments: \( \mathcal{A}(x, y) \)

- The probability of translation \( y \) to \( x \) through an alignment \( a \) is \( \Pr(x, a | y) \)

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

\[
\Pr(x, a | y) = \Pr(J | y) \cdot \Pr(x, a | J, y)
\]

\[
= \Pr(J | y) \cdot \Pr(a | J, y) \cdot \Pr(x | a, J, y)
\]

- **Length probability**: \( \Pr(J | y) \)
- **Alignment probability**: \( \Pr(a | J, y) \)
- **Lexicon probability**: \( \Pr(x | a, J, y) \)

\[
\Pr(a | J, y) = \prod_{j=1}^{J} \Pr(a_j | a_i^{j-1}, J, y) \quad \Pr(x | a, J, y) = \prod_{j=1}^{J} \Pr(x_j | x_i^{j-1}, a, J, y)
\]

\[
\Pr(x, a | y) = \Pr(J | y) \cdot \prod_{j=1}^{J} \Pr(a_j | a_i^{j-1}, x_i^{j-1}, J, y) \cdot \Pr(x_j | a_i^{j}, x_i^{j-1}, J, y)
\]
Index

1 Statistical framework to machine translation ▷ 2

2 Alignments ▷ 11

○ 3 Statistical alignment models ▷ 20

4 First-order alignment models ▷ 50

5 Categorization in statistical modeling ▷ 66

6 Bibliography ▷ 74

An inverse approach

\[
\text{SEARCH} \quad \underset{y}{\text{argmax}} \quad P_r(x|y) \cdot P_r(y)
\]

\[
P_r(x | y)
\]

\[
P_r(y)
\]

ALIGNMENT MODELS AND LEXICON MODELS

BILINGUAL TRAINING DATA

TARGET LANGUAGE MODEL

TARGET TRAINING DATA
**Pattern Recognition approaches to Machine Translation**  

**Statistical Alignment Models**

- **Zero-order models**
  - Model 1
  - Model 2
  - The Viterbi approximation
  - The search problem

---

**Model 1**

\[
\Pr(x, a \mid y) = \Pr(J \mid y) \cdot \prod_{j=1}^{J} \Pr(a_j \mid a_{i-1}^j, x_{i-1}^j, J, y) \cdot \Pr(x_j \mid a_{i}^j, x_{i-1}^j, J, y)
\]

- \(\Pr(J \mid y) \approx n(J \mid I)\)
- \(\Pr(a_j \mid a_{i-1}^j, x_{i-1}^j, J, y) \approx \frac{1}{(I+1)^j}\)
- \(\Pr(x_j \mid a_{i}^j, x_{i-1}^j, J, y) \approx l(x_j \mid y_i)\)

\(l(x_j \mid y_i)\) defines a **statistical lexicon**

\[
\Pr(x \mid y) \approx PM_1(x \mid y) = \frac{n(J \mid I)}{(I + 1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} l(x_j \mid y_i)
\]
Pattern Recognition approaches to Machine Translation

Statistical Alignment Models

Model 1

\[
\Pr(x \mid y) = \sum_a \Pr(J \mid y) \cdot \Pr(x, a \mid J, y)
\]

\[
\approx \sum_a n(J \mid I) \cdot \prod_{j=1}^{J} \left[ \frac{1}{(I+1)^J} \cdot l(x_j \mid y_{a_j}) \right]
\]

\[
= \frac{n(J \mid I)}{(I+1)^J} \sum_{a_1=0}^{I} \cdots \sum_{a_J=0}^{I} \prod_{j=1}^{J} l(x_j \mid y_{a_j})
\]

\[
= \frac{n(J \mid I)}{(I+1)^J} \prod_{j=1}^{J} \sum_{a_j=0}^{I} l(x_j \mid y_{a_j})
\]

\[
= \frac{n(J \mid I)}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} l(x_j \mid y_i) = P_{M1}(x \mid y)
\]

Generative process: Given a target sentence \(y\) of length \(I\),

1. Choose the length of the source sentence \(J\) according to \(n(J \mid I)\)

2. For each \(1 \leq j \leq J\), choose a position \(a_j\) in the target sentence according to an uniform distribution.

3. For each \(1 \leq j \leq J\) choose a source word \(x_j\) according to \(l(x_j \mid y_{a_j})\)
### Parameter estimation with Model 1

- **Training sample:** \( A = \{(x^{(1)}_1, y^{(1)}_1), (x^{(2)}_2, y^{(2)}_2), \ldots, (x^{(K)}_K, y^{(K)}_K)\} \)

- **Function to be maximized:** likelihood

\[
\mathcal{L}_A(l) = \prod_{k=1}^{K} P_{M1}(x^{(k)} | y^{(k)}) = \prod_{k=1}^{K} \frac{n(J^{(k)} | I^{(k)})}{I^{(k)} + 1} \frac{J^{(k)} I^{(k)}}{{J^{(k)} I^{(k)}}} \prod_{j=1}^{I^{(k)}} \sum_{i=0}^{l(x^{(k)}_j | y^{(k)}_i)} l(x^{(k)}_j | y^{(k)}_i)
\]

or the log-likelihood

\[
\mathcal{L}_A(l) = \sum_{k=1}^{K} \sum_{j=1}^{J^{(k)}} \log \sum_{i=0}^{I^{(k)}} l(x^{(k)}_j | y^{(k)}_i)
\]

- **Procedure:** Expectation-maximization or growth transformations \((T_1 : \theta \rightarrow \theta)\):

\[
T_1(l(x | y)) = \frac{l(x | y) \cdot \left( \frac{\partial \mathcal{L}_A(l)}{\partial l(x | y)} \right)}{\sum_{x'} l(x' | y) \cdot \left( \frac{\partial \mathcal{L}_A(l)}{\partial l(x' | y)} \right)}
\]
Parameter estimation with Model 1

\[ l(x \mid y) \frac{\partial L_A(l)}{\partial l(x \mid y)} = l(x \mid y) \sum_{k=1}^{K} \prod_{k' \neq k} P_{M1}(x^{(k')} \mid y^{(k')}) \frac{\partial P_{M1}(x^{(k)} \mid y^{(k)})}{\partial l(x \mid y)} = L_A(l) \sum_{k=1}^{K} \frac{l(x \mid y) \cdot \frac{\partial P_{M1}(x^{(k)} \mid y^{(k)})}{\partial l(x \mid y)}}{P_{M1}(x^{(k)} \mid y^{(k)})} \]

\[ = l(x \mid y) \cdot \frac{n(J^{(k)} \mid I^{(k)})}{(I^{(k)} + 1)^{j^{(k)}}} \sum_{j=1}^{j^{(k)}} \left( \sum_{i=0}^{l^{(k)}} \delta(x^{(k)}_i, x) \cdot \delta(y^{(k)}_i, y) \right) \]

Parameter estimation in Model 1

Iterative E-M procedure:

**Expectation step:**

\[ c(x \mid y; x^{(k)}, y^{(k)}) = \frac{l(x \mid y)}{\sum_{i=0}^{l^{(k)}} l(x \mid y^{(k)}_i)} \cdot \#(y, y^{(k)}) \cdot \#(x, x^{(k)}) \]

**Maximization step:**

\[ T_1(l(x \mid y)) = \frac{\sum_{k=1}^{K} c(x \mid y; x^{(k)}, y^{(k)})}{\sum_{x'} \sum_{k=1}^{K} c(x' \mid y; x^{(k)}, y^{(k)})} \]
Pattern Recognition approaches to Machine Translation

Statistical Alignment Models

Parameter estimation in Model 1

- **PROPERTY**: the increase of the likelihood of the training set in each iteration:

\[
\prod_{k=1}^{K} P_{M1}(x^{(k)} | y^{(k)}) \leq \prod_{k=1}^{K} P_{T_1(M1)}(x^{(k)} | y^{(k)})
\]

- **PROPERTY**: Eventually an absolute maximum is achieved!

- **COMPUTATIONAL COST OF T_1**: If \( I_M = \max_k I^{(k)} \) and \( J_M = \max_k J^{(k)} \)

  - time: \( O(K \times (I_M + J_M)) \)
  - space: \( O(|\Sigma| \times |\Delta|) \)

Model 2

\[
Pr(x, a | y) = Pr(J | y) \cdot \prod_{j=1}^{J} Pr(a_j | a_{j-1}^{j-1}, x_{j-1}^{j-1}, J, y) \cdot Pr(x_j | a_j^{j-1}, y)
\]

- \( Pr(J | y) \approx n(J|I) \)
- \( Pr(a_j | a_{j-1}^{j-1}, x_{j-1}^{j-1}, J, y) \approx a(a_j | j, J, I) \)
- \( Pr(x_j | a_j^{j-1}, J, y) \approx l(x_j | y_{a_j}) \)

\( l(x_j | y_{a_j}) \) defines a **statistical lexicon**

\( a(i | j, J, I) \) defines **statistical alignments**

\[
Pr(x | y) \approx P_{M2}(x | y) = n(J|I) \cdot \prod_{j=1}^{J} \sum_{i=0}^{I} a(i | j, J, I) \cdot l(x_j | y_i)
\]
Pattern Recognition approaches to Machine Translation  
Statistical Alignment Models

Model 2

\[
\Pr(x | y) = \sum_a \Pr(J | y) \cdot \Pr(x, a | J, y)
\]

\[
= \sum_a n(J | I) \cdot \prod_{j=1}^J \left[ a(a_j | j, J, I) \cdot l(x_j | y_{a_j}) \right]
\]

\[
= n(J | I) \cdot \sum_{a_1=0}^I \cdots \sum_{a_J=0}^I \prod_{j=1}^J \left[ a(a_j | j, J, I) \cdot l(x_j | y_{a_j}) \right]
\]

\[
= n(J | I) \cdot \prod_{j=1}^J \sum_{a_j=0}^I a(a_j | j, J, I) \cdot l(x_j | y_{a_j})
\]

\[
= n(J | I) \cdot \prod_{j=1}^J \sum_{i=0}^I a(i | j, J, I) \cdot l(x_j | y_i) = P_{M2}(x | y)
\]

Generative process: Given a target sentence \( y \) of length \( I \),

1. Choose the length of the source sentence \( J \) according to \( n(J | I) \).
2. For each \( 1 \leq j \leq J \), choose a position \( a_j \) in the target sentence according to \( a(a_j | j, J, I) \).
3. For each \( 1 \leq j \leq J \) choose a source word \( x_j \) according to \( l(x_j | y_{a_j}) \).
An example

Given $y$: a double room ($I = 3$)

<table>
<thead>
<tr>
<th>Choose $J$ ($n(J \mid 3)$): ($J = 5$)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose $a_j$ ($a(a_j \mid j, I, J)$)</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Choose $x_j$ ($l(x_j \mid y_i)$)</td>
<td>Una habitación con dos camas</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameter estimation in Model 2

- Training sample: $A = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(K)}, y^{(K)})\}$
- Function to be maximized: likelihood

$$
\mathcal{L}_A(a, l) = \prod_{k=1}^{K} P_{M2}(x^{(k)} \mid y^{(k)})
= \prod_{k=1}^{K} n(J^{(k)} \mid I^{(k)}) \cdot \prod_{j=1}^{J} \sum_{i=0}^{I} a(i \mid j, J^{(k)}, I^{(k)}) \cdot l(x_j^{(k)} \mid y_i^{(k)})
$$

or the log-likelihood:

$$
\mathcal{L}_A(a, l) = \sum_{k=1}^{K} \sum_{j=1}^{J} \sum_{i=0}^{I} \log a(i \mid j, J^{(k)}, I^{(k)}) \cdot l(x_j^{(k)} \mid y_i^{(k)})
$$

- Procedure: Expectation-maximization or growth transformations ($T_2 : \theta \rightarrow \theta$)
Parameter estimation in Model 2

Iterative E-M procedure:

### Expectation step:

\[ c(x \mid y; x^{(k)}, y^{(k)}) = \sum_{j=1}^{J^{(k)}} \sum_{i=0}^{I^{(k)}} \frac{l(x \mid y) \cdot a(i \mid j; x^{(k)}, y^{(k)}) \cdot \delta(x, x^{(k)}_j) \cdot \delta(y, y^{(k)}_i)}{\sum_{n=0}^{I^{(k)}} l(x \mid y_n) \cdot a(n \mid j, I^{(k)})} \]

\[ c(i \mid j; I, x^{(k)}, y^{(k)}) = \begin{cases} \frac{l(x^{(k)}_j \mid y^{(k)}_i) \cdot a(i \mid j; x^{(k)}, y^{(k)})}{\sum_{i' = 0}^{I^{(k)}} l(x^{(k)}_j \mid y^{(k)}_{i'}) \cdot a(i' \mid j; x^{(k)}, y^{(k)})} & \text{if } I = I^{(k)} \\
0 & \text{otherwise} \end{cases} \]

### Maximization step:

\[ T_2(l(x \mid y)) = \frac{\sum_{k=1}^{K} c(x \mid y; x^{(k)}, y^{(k)})}{\sum_{x'} \sum_{k=1}^{K} c(x' \mid y; x^{(k)}, y^{(k)})} \]

\[ T_2(a(i \mid j; I, x^{(k)}, y^{(k)})) = \frac{\sum_{k=1}^{K} c(i \mid j; I, x^{(k)}, y^{(k)})}{\sum_{i'} \sum_{k=1}^{K} c(i' \mid j; I, x^{(k)}, y^{(k)})} \]

F. Casacuberta – DSIC-ITI-UPV
24-28 January 2005

- **PROPERTY**: the increase of the likelihood of the training set in each iteration.

\[ \prod_{k=1}^{K} P_{M2}(x^{(k)} \mid y^{(k)}) \leq \prod_{k=1}^{K} P_{T_2(M2)}(x^{(k)} \mid y^{(k)}) \]

- **PROPERTY**: Eventually an local maximum is achieved.

- **COMPUTATIONAL COST OF $T_2$**: If $I_M = \max_k I^{(k)} \quad J_M = \max_k J^{(k)}$
- time: $O(K \times I_M \times J_M)$
- space: $O((|\Sigma| \times |\Delta|) + I_M + J_M)$
Optimal alignment with Model 2

\[ P_{M2}(x | y) = n(J | I) \cdot \prod_{j=1}^{J} \sum_{i=0}^{I} a(i | j, J, I) \cdot l(x_j | y_i) \approx \]

\[ \hat{P}_{M2}(x | y) = n(J | I) \cdot \prod_{j=1}^{J} \max_{0 \leq i \leq I} a(i | j, I, J) \cdot l(x_j | y_i) \]

Algorithm Viterbi \((x, y, l, a)\)

**Input:** A pair \(x, y\) and the parameters \(l\) and \(a\) of Model 2

**Output:** An optimal alignment \(A\) between \(x\) and \(y\).

For \(j := 1\) until \(J\)

\[
A[j] := \arg\max_{0 \leq i \leq I} a(i | j, J, I) \cdot l(x_j | y_i)
\]

End-for

Return: \(A\)

The computational cost of this algorithm is \(O(J \times I)\).

Examples of alignments

EUTRANS-I corpus: Spanish-English

- **Vocabulary:** 680 Spanish words, and 513 English words.

- **Training:** 10,000 pairs (97,000/99,000 words).

An example

1 2 3 4 5 6 7 8 9 10

por favor , ¿ podría ver alguna habitación tranquila ?

- **MODEL 1, ITERATION 5**
  could (5) I (6) see (6) a (7) quiet (9) room (8) , (3) please (2) ? (4)

- **MODEL 2, ITERATION 2**
  could (5) I (6) see (6) a (7) quiet (9) room (8) , (3) please (3) ? (10)
Examples of alignments

**MODEL 2 ITERATION 2**

I have made a reservation for Federico Redondo.

Could you ask for our taxi, please?

Would you mind waking us up tomorrow at a quarter past seven, please?

I am leaving on Thursday June the third at half past one in the afternoon.

---

**Viterbi estimation**

- **Training sample:** \( A = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(K)}, y^{(K)})\} \)
- **Function to be maximized:** Viterbi score
- **Procedure:**

\[
\hat{c}(x | y; x^{(k)}, y^{(k)}) = \#(x, x^{(k)}) \times \#(y, y^{(k)})
\]

\[
\hat{c}(i | j; J, I, x^{(k)}, y^{(k)}) = \begin{cases} 
1 & \text{if } i = a_j \text{ in VITERBI}(x^{(k)}, y^{(k)}, l, a) \text{ and if } J^{(k)} = J \text{ and } I^{(k)} = I \\
0 & \text{otherwise}
\end{cases}
\]

\[
T_v(l(x | y)) = \frac{\sum_{k=1}^{K} \hat{c}(x | y; x^{(k)}, y^{(k)})}{\sum_{x'} \sum_{k=1}^{K} \hat{c}(x' | y; x^{(k)}, y^{(k)})}
\]

\[
T_v(a(i | j, J, I)) = \frac{\sum_{k=1}^{K} \hat{c}(i | j; J, I, x^{(k)}, y^{(k)})}{\sum_{i'} \sum_{k=1}^{K} \hat{c}(i' | j; J, I, x^{(k)}, y^{(k)})}
\]
Viterbi estimation

- PROPERTY: the increasing of the Viterbi score in each iteration:

\[
\prod_{k=1}^{K} \hat{P}_{M2}(x^{(k)} \mid y^{(k)}) \leq \prod_{k=1}^{K} \hat{P}_{T_\nu(M2)}(x^{(k)} \mid y^{(k)})
\]

- PROPERTY: Eventually a local maximum is achieved.

- COMPUTATIONAL COST OF \( T_\nu \): If \( I_M = \max_k I^{(k)} \) and \( J_M = \max_k J^{(k)} \)
  - time: \( O(K \times I_M \times J_M) \)
  - space: \( O((|\Sigma| \times |\Delta|) + I_M \times J_M) \)

Simplified version of Model 2 (Ney et al. 2000)

\[
\Pr(x, a \mid y) = \Pr(J \mid y) \cdot \prod_{j=1}^{J} \Pr(a_j \mid a_{i-1}^j, x_{i-1}^j, J, y) \cdot \Pr(x_j \mid a_i^j, x_{i-1}^j, J, y)
\]

- \( \Pr(J \mid y) \approx n(J \mid I) \)
- \( \Pr(a_j \mid a_{i-1}^j, x_{i-1}^j, J, y) \approx a(a_j \mid j, I) \)
- \( \Pr(x_j \mid a_i^j, x_{i-1}^j, J, y) \approx l(x_j \mid y_a) \)

\[
\Pr(x \mid y) \approx P^A(x \mid y) = n(J \mid I) \cdot \prod_{j=1}^{J} \sum_{i=1}^{I} a(i \mid j, I) \cdot l(x_j \mid y_i)
\]

\[
T_A(t(x \mid y)) = \frac{\sum_{k=1}^{K} c(x \mid y; x^{(k)}, y^{(k)})}{\sum_{x'} \sum_{k=1}^{K} c(x' \mid y; x^{(k)}, y^{(k)})} \text{ and } T_A(a(i \mid j, J)) = \frac{r(i - j \frac{1}{2})}{\sum_{i'=1}^{I} r(i' - j \frac{1}{2})}
\]

It is assumed that the diagonal of the plain \((j,i)\) is the dominant factor.
The translation process: searching

$$\text{argmax}_{y} \Pr(x \mid y) \cdot \Pr(y)$$

A computational difficult problem
(K.Knight Decoding complexity in word-replacement translation models. Comp. Ling. 1999)

Algorithmic solutions:
- Dynamic Programming like (Garcia-Varea, 1998) (Ney, 2000)
- Stack-Decoding, A* or Branch & Bound (Brown, 1990) (Wang, 1997)
- Greedy (Germann, 2001)
- Using finite-state transducers (Kumar, 2004)

Dynamic-programming like search

An approximate solution: DPSearchM2

- Characteristics:
  - Building partial hypothesis (1, 2, ⋅⋅⋅ , I)
  - Search graph: a $|\mathcal{E}| \times I$ trellis.

- Assumptions:
  - Language model: $n$-grams (bigrams)
  - The length of target sentence is known: $I$. 

F. Casacuberta – DSIC-ITI-UPV 24-28 January 2005 2:44
Dynamic-programming like search: \textit{DPSearchM2}

- Search criterium:
  \[
  \max_y \left\{ \prod_{i=1}^{I} p_2(y_i|y_{i-1}) \cdot \prod_{j=1}^{J} \sum_{i=0}^{I} l(x_j|y_i) \cdot a(i|j, J, I) \right\}
  \]

- Auxiliar variables:
  - \(Q(e, i, j)\): Contribution of the translation models for each \(j\):
    \[
    Q(t, i, j) = l(x_j|t) \cdot a(i|j, J, I) + \sum_{k=0}^{i-1} l(x_j|y_k) \cdot a(k|j, J, I)
    \]
  - \(T(t, i)\): Contribution of the language model:
    \[
    T(t, i) = p_2(t|y_{i-1}) \prod_{k=1}^{i-1} p(y_k|y_{k-1})
    \]
Dynamic-programming like search: **DPSearchM2**

- **Recursion:**
  \[
  Q(t, i, j) = Q(\hat{t}, i - 1, j) + l(x_j|\hat{t}) \cdot a(i|j, J, I) \\
  T(t, i) = T(\hat{t}, i - 1) \cdot p_2(t|\hat{t})
  \]

  where \( \hat{t} \) is the optimal state in \( i - 1 \)

  \[
  \hat{t} = \arg\max_{t'} T(t', i - 1) \cdot p_2(t' \mid t') \cdot \prod_{j=1}^{J} (Q(t', i - 1, j) + l(x_j|t) \cdot a(i|j, J, I))
  \]

- Basis of the recursion \( \forall t \ \forall j : 1 \leq j \leq J \):
  \[
  Q(t, 1, j) = l(x_j|y_0) \cdot a(0|j, J, I) \\
  T(t, 1) = 1.0
  \]

- An approximation to the optimal solution is:
  \[
  \hat{y} = \arg\max_{y} \left\{ T(y_I|I) \cdot \prod_{j=1}^{J} Q(y_I, I, j) \right\}
  \]

- **Problem:** in \( i, y^f_{i+1} \) is unknown:

  \[
  Q(t, i, j) = l(x_j|t) \cdot a(i|j, J, I) + \sum_{k=0; k \neq i}^{J} l(x_j|y_k) \cdot a(k|j, J, I)
  \]

- **Solution:** Iterative search

  \[
  Q(t, i, j) = Q(\hat{t}(t, i), i - 1, j) + l(x_j|\hat{t}(t, i)) \cdot a(i|j, J, I) + R(j, i + 1)
  \]

  \[
  R(j, i) = \sum_{k=1}^{I} l(x_j|\bar{y}_k) a(k|j, J, I); \quad \bar{y}^f_1 \text{ is the last optimal solution}
  \]

  - **Initialization:** \( R(j, i) = 0 \)
  - **But**

    \[
    R(i, j) = \sum_{k=i}^{I} \max_t \{ l(x_j|t) \cdot a(k|j, J, I) \} \rightarrow \text{Heuristic initialization}
    \]
First-order models

- Homogeneous HMM model (HMM)
- Search: Quasi-monotone alignments
- Search: Inverted alignments
- Results
- Other search solution
Homogeneous HMM alignment

\begin{align*}
\Pr(x, a \mid y) &= \Pr(J \mid y) \cdot \prod_{j=1}^{J} \Pr(a_j \mid a_{j-1}^i, x_{j-1}^i, J, y) \cdot \Pr(x_j \mid a_j^i, x_{j-1}^i, J, y)
\end{align*}

- \Pr(J \mid y) \approx n(J \mid I)
- \Pr(a_j \mid a_{j-1}^i, x_{j-1}^i, J, y) \approx h(a_j \mid a_{j-1}, J, I)
- \Pr(x_j \mid a_j^i, x_{j-1}^i, J, y) \approx l(x_j \mid y_{a_j})

\begin{align*}
h(a_j \mid a_{j-1}, J, I) \text{ defines statistical alignment with first-order dependencies}
\end{align*}

\begin{align*}
P_{HMM}(x \mid y) &= n(J \mid I) \cdot \sum_{a} \prod_{j=1}^{J} h(a_j \mid a_{j-1}, J, I) \cdot l(x_j \mid y_{a_j})
\end{align*}

\begin{align*}
&\text{Forward computation of } P_{HMM}(x \mid y) \\
&P_{HMM}(x \mid y) \approx n(J \mid I) \cdot \sum_{a} \prod_{j=1}^{J} h(a_j \mid a_{j-1}, J, I) \cdot l(x_j \mid y_{a_j}) = n(J \mid I) \cdot Q(I, J)
\end{align*}

\begin{align*}
Q(i, j) &= l(x_j \mid y_i) \cdot \sum_{i'} h(i \mid i', J, I) \cdot Q(i', j - 1)
\end{align*}

\begin{align*}
\text{MAXIMUM APPROACH}
&P_{HMM}(x \mid y) \approx n(J \mid I) \cdot \max_{a} \prod_{j=1}^{J} h(a_j \mid a_{j-1}, J, I) \cdot l(x_j \mid y_{a_j})
\end{align*}

\begin{align*}
\text{Viterbi computation in the maximum approach}
\hat{Q}(i, j) &= l(x_j \mid y_i) \cdot \max_{i'} \left( h(i \mid i', J, I) \cdot \hat{Q}(i', j - 1) \right)
\end{align*}
Homogeneous HMM alignment

ALIGNMENT PROBABILITY DISTRIBUTION:

\[ h(i|i', I, J) = \frac{q(i - i')}{\sum_{i''=1}^{I} q(i'' - i')} \]

TRAINING WITH THE MAXIMUM APPROACH

- Position alignment by computing \( \hat{Q}(i, j) \)
- Parameter estimation (relative frequencies)

Searching with homogeneous HMM alignments

\[
\max_y \Pr(y) \cdot \Pr(x \mid y) \approx \max_I \left\{ n(J \mid I) \cdot \max_{y \in \Delta^I} \left( \prod_{i=1}^{I} p_2(y_i \mid y_{i-1}) \cdot \max_a \prod_{j=1}^{J} \left[ h(a_j \mid a_{j-1}, J) \cdot l(x_j \mid y_{a_j}) \right] \right) \right\}
\]

- Quasi-monotone alignments and quasi-monotone search.
  \[ p_2(y_i \mid y_{i-1}) \Rightarrow p[a_j - a_{j-1}](y_{a_j} \mid y_{a_{j-1}}) \]
- Inverted alignments and search.
  \[ h(a_j \mid a_{j-1}, J) \cdot l(x_j \mid y_{a_j}) \Rightarrow q(i \mid b_i, J, I) \cdot t(x_{b_i} \mid y_i) \]
Quasi-monotone alignments and quasi-monotone search

\[ \delta \equiv a_j - a_{j-1} \in \{0, 1, 2\} \]

Modification of the target language model

- If \( \delta = 0 \), \( p_{(\delta)}(y \mid y') = \begin{cases} 1 & y = y' \\ 0 & y \neq y' \end{cases} \)
- If \( \delta = 1 \), \( p_{(\delta)}(y \mid y') = p_2(y \mid y') \)
- If \( \delta = 2 \), \( p_{(\delta)}(y \mid y') = \max_{y''} (p_2(y \mid y'') \cdot p_2(y'' \mid y')) \)

\[
\max \left\{ n(J \mid I) \cdot \max_{y, a} \left\{ \prod_{j=1}^{J} h(a_j \mid a_{j-1}, J) \cdot p_{a_j-a_{j-1}}(y_{a_j} \mid y_{a_{j-1}}) \cdot l(x_j \mid y_{a_j}) \right\} \right\}
\]
Quasi-monotone alignments and quasi-monotone search

\[
\max_I \left\{ n(J \mid I) \cdot \max_{y,a} \left\{ \prod_{j=1}^{J} h(a_j \mid a_{j-1}, J) \cdot p[a_j-a_{j-1}] (y_{a_j} \mid y_{a_{j-1}}) \cdot l(x_j \mid y_{a_j}) \right\} \right\}
\]

\(Q(i, j, s)\) = the probability of the best partial hypothesis \((y_i^1, a_j^1)\) with \(y_i = y\) and \(a_j = i\).

\[
Q(i, j, s) = t(x_j \mid s) \cdot \max_{\delta, e'} \left\{ h(i - \delta, I) \cdot p[\delta] (s \mid s') \cdot Q(i - \delta, j - 1, s') \right\}
\]

Solution

\[
\max_{I, \hat{s}} (n(J \mid I) \cdot Q(I, J, \hat{s}))
\]

Computational cost: \(O(I_{max} \cdot J \cdot |\Delta|^2)\)

Problem with the monotone models: assumption of similar syntactic structures in both languages.

- First solution: Re-ordering and monotone models.
- Second solution: Two-level monotone alignments:
- Third solution: a new alignment model
  * Concept of inverted alignment: \(b_i = j \Rightarrow y_i \leftrightarrow x_j\)
  * Associated distribution: \(q(i \mid b_i, J, I)\)
  * (+ optional) Fertility
An inverted alignment is: $i \rightarrow B_i \subset \{1, \ldots, j, \ldots, J\}$.

$$\Pr(x, b \mid y) = \Pr(J \mid y) \cdot \Pr(x, b \mid J, y)$$

$$= \Pr(J \mid y) \cdot \prod_{i=1}^{I} \Pr(x_{b_i}, b_i \mid x_{b_1}^{b_i-1}, b_1^{i-1}, J, y)$$

$$= \Pr(J \mid y) \cdot \prod_{i=1}^{I} \left( \Pr(b_i \mid x_{b_1}^{b_i-1}, b_1^{i-1}, J, y) \cdot \Pr(x_{b_i} \mid x_{b_1}^{b_i-1}, b_1^{i-1}, J, y) \right)$$

$$\approx n(J \mid I) \cdot \prod_{i=1}^{I} \left( q(b_i \mid b_1^{i-1}; x_{b_i}, y_{i-1}) \cdot l(x_{b_i} \mid y_i) \right)$$

$$= n(J \mid I) \cdot \prod_{i=1}^{I} \left( q(b_i \mid b_1^{i-1}; x_{b_i}, y_{i-1}) \cdot \prod_{j \in b_i} l(x_j \mid y_i) \right)$$

**Inverted alignments and search**

$$\max_I \left\{ n(J \mid I) \cdot \max_{y,b} \left\{ \prod_{i=1}^{I} \left[ p_2(y_i \mid y_{i-1}) \cdot q(i \mid b_i, J, I) \cdot l(x_{b_i} \mid y_i) \right] \right\} \right\}$$

$Q_I(i, j, y) = \text{probability of the best partial hypothesis } (y_1^i, b_1^i) \text{ con } y_i = y \text{ y } b_i = j$.

**General recursion:**

$$Q_I(i, j, y) = l(x_j \mid y) \cdot q(i \mid j, I, J) \cdot \max_{j',y'} (p_2(y \mid y') \cdot Q_I(i - 1, j', y'))$$

**Solution:**

$$\max_I (n(J \mid I) \cdot Q_I(I, J, \hat{y}))$$

**Computational cost:** $O(I_{\text{max}}^2 \cdot J^2 \cdot |\Delta|^2)$
Inverted alignments


Results

EuTrans-I corpus (Spanish-English)

- Vocabulary: 680 Spanish words, and 513 English words.
- Training: 10,000 pairs (97,000/99,000 words).
- Test: 2,996 pairs (PP=8.6/5.2) (35,000/35,590 words).

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quasi-monotone search</td>
<td>10.8</td>
</tr>
<tr>
<td>DP-search with M2</td>
<td>13.9</td>
</tr>
</tbody>
</table>

Word error rate (WER): The minimum number of substitution, insertion and deletion operations needed to convert the word string hypothesized by the translation system into a given single reference word string.
Dynamic programming approach: The traveling salesman problem

DP Algorithm for SMT \((x, l, a)\)

Input: A source sentence \(x\) and the parameters \(l\) and \(a\).
Output: A target sentence \(y\).

Initialization

For \(c := 1\) until \(J\) do

For each \((C, j)\) with \(C \subset \{1, \ldots, J\}\), \(j \in C\) and \(|C| = c\) do

For each pair of target words \(y, y'\) do

\[
Q_{yy'}(C, j) = l(x_j \mid y) \cdot \max_{y'', j' \in C - \{j\}} (a(j \mid j') \cdot p(y \mid y', y'') \cdot Q_{y'y''}(C - \{j\}, j'))
\]

End-for

End-for

Return: \(\arg\max_{y, y', j} p(\# \mid y, y', j) \cdot Q_{yy'}(\{1, \ldots, J\}, j)\) and traceback

The set \(C\) is constraint to be a maximum number of positions.
Index

1 Statistical framework to machine translation ▷ 2
2 Alignments ▷ 11
3 Statistical alignment models ▷ 20
4 First-order alignment models ▷ 50
5 Categorization in statistical modeling ▷ 66
6 Bibliography ▷ 74

Categorization

• Too many parameters to be estimated

• Many words play the same role: names, dates, etc.

• Substitution of words by categories:
  – The vocabulary size decreases.
  – Easy word addition to the vocabulary.

• Examples:
  – mi nombre es $NAME.masc $SURNAME . # my name is $NAME.masc $SURNAME .
  – nos vamos a ir el $DATE a $HOUR . # we are leaving on $DATE at $HOUR .

• Given a bilingual corpus:
  – Automatic extraction of bilingual categories.
  – Manual extraction of bilingual categories.
Categorization and learning

- Given a bilingual corpus:
  - **CATEGORIZED TRANSLATOR**: Training a statistical translator (a translation model plus a target language model) from a corpus of categorized pairs.
  - **A TRANSLATOR FOR EACH CATEGORY**: Training a statistical translator (translation model plus target language model) from the set of pairs of segments associated to each category.
  - **A SOURCE CATEGORIZER**: Training a statistical translator (translation model plus language model) from the set of source no-categorized/categorized sentences.

### An approach


1. **CATEGORIZATION**: Translating the source sentence into an source categorized sentence and Obtaining the source instances of each category.
2. **CATEGORIZED TRANSLATION**: Translating the source categorized sentence into a target categorized sentence.
3. **TRANSLATION OF EACH CATEGORY**: Translating the source instances of each category detected.
4. **CATEGORY RESOLUTION**: Substitution of each target category by the corresponding instance translation.
An example


me voy a ir el día veintiséis de abril a las doce en punto de la mañana

Statistical Categorization
$\text{DATE = veintiséis de abril}$
$\text{HOUR = las doce en punto}$

Viterbi Alignment

I am leaving on $\text{DATE}$ at $\text{HOUR}$ in the morning

Statistical Translation
$\text{DATE = April the twenty-sixth}$
$\text{HOUR = twelve o’clock}$

Category Resolution

I am leaving on April the twenty-sixth at twelve o’clock in the morning

Another approach

1. **Categorization**: Translating the source sentence into an source categorized sentence.
2. **Categorized Translation**: Translating the source categorized sentence into a target categorized sentence.
3. **Detailed Translation**: Translating the source non-categorized sentence using the target categorized sentence as a restricted target language.
Results
EuTrans-I corpus (Spanish-English)

- Vocabulary: 680 Spanish words, and 513 English words.
- Training: 10,000 pairs (97,000/99,000 words).
- Test: 2,996 pairs (PP=8.6/5.2) (35,000/35,590 words).

<table>
<thead>
<tr>
<th>Model</th>
<th>categorization</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quasi-monotone search</td>
<td>manual</td>
<td>6.7</td>
</tr>
<tr>
<td>DP-search with M2</td>
<td>manual</td>
<td>9.8</td>
</tr>
<tr>
<td>Quasi-monotone search</td>
<td>no</td>
<td>10.8</td>
</tr>
<tr>
<td>DP-search with M2</td>
<td>no</td>
<td>13.9</td>
</tr>
</tbody>
</table>

Automatic categorization

- **Extended word categories**
  (Barrachina & Vilar. *Bilingual clustering using monolingual algorithms*. TMI. 1999.)
  1. Align a bilingual corpus
  2. Build extended words using the alignments
  3. Apply a clustering algorithm to the corpus of extended word sentences

- **Statistical bilingual categories**
  (Och. *An Efficient method for determining bilingual word classes*. ECACL. 1999.)
  1. Align a bilingual corpus
  2. Apply a clustering algorithm to the target corpus.
  3. Apply a clustering algorithm to the source corpus taking into account the categories of target words aligned to the source words.
Index

1. Statistical framework to machine translation  ▷  2
2. Alignments  ▷  11
3. Statistical alignment models  ▷  20
4. First-order alignment models  ▷  50
5. Categorization in statistical modeling  ▷  66
6. Bibliography  ▷  74

Bibliography

3: Advanced Statistical Alignment Models

Francisco Casacuberta Nolla

fcn@iti.upv.es

24-28 January 2005

Index

1 Statistical framework to machine translation ▷ 2

2 Fertility-based models ▷ 7

3 The search problem ▷ 27

4 Maximum entropy models ▷ 42

5 Using linguistic knowledge ▷ 56

6 Bibliography ▷ 65
**Index**

- 1. *Statistical framework to machine translation* ▷ 2
- 2. Fertility-based models ▷ 7
- 3. The search problem ▷ 27
- 4. Maximum entropy models ▷ 42
- 5. Using linguistic knowledge ▷ 56
- 6. Bibliography ▷ 65

**An inverse approach**

\[
\text{SEARCH } \arg\max_y P_r(x|y) \cdot P_r(y)
\]

\[
P_r(x \mid y) \quad P_r(y)
\]

ALIGNMENT MODELS AND LEXICON MODELS

BILINGUAL TRAINING DATA

TARGET LANGUAGE MODEL

TARGET TRAINING DATA
An example of word alignments

I have made a reservation for Rosario Cabedo.

Alignments

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

Alignment probabilities and lexicon probabilities

- Model 1
- Model 2
- Hidden Markov model
Pattern Recognition approaches to Machine Translation

Advanced Statistical Alignment Models

Models 1, 2 or HMM

Index

1 Statistical framework to machine translation ▷ 2

2 Fertility-based models ▷ 7

3 The search problem ▷ 27

4 Maximum entropy models ▷ 42

5 Using linguistic knowledge ▷ 56

6 Bibliography ▷ 65
Fertility-based models

- Fertility
- Model 3
- Model 4
- Model 5
- Model 6
- The training process

Models 3, 4, 5 and 6

- Model 3 is a zero-order model: Lexicon, fertility and distortion models.
- Model 4 is a refined version (first-order) of distortion distribution in Model 3.
- Model 5 is a consistent version of distortion distribution in Model 4.
- Model 6 is a log-linear combination of HMM and Model 4.
Fertility

**Fertility** $\phi$ of $y_i \in \Delta$: number of the source words connected to a target word $y_i$.

1. Choose how many source words are connected to a target word $y_i$: *fertility* of $y_i$

   $$\Phi = \phi(y_i)$$

2. Choose a set of the source words, a *tablet* $\tau_i$, that is connected to the $i$-th target word

   $$\Gamma_{i,k} = \tau_{i,k} \in \Sigma \text{ for } 1 \leq k \leq \phi(y_i)$$

3. Choose the *position* $\pi_{i,k}$ in the source sentence of the $k$-th word $\tau_{i,k}$ that is connected to the $i$-th target word

   $$\Pi_{i,k} = \pi_{i,k}, 1 \leq \pi_{i,k} \leq J$$
An example

Given \( y: \) a double room \( (I = 3) \)

\[
\begin{array}{l|llll}
  i & 1 & 2 & 3 \\
\hline
\text{Choose } \phi(y_i) = \phi & 1 & 3 & 1 \\
\text{Choose } \tau_{i,k} = x & \{\text{una}\} & \{\text{con, camas, dos}\} & \{\text{habitación}\} \\
\text{Choose } \pi_{i,k} = j & 1 & 3 & 5 & 4 & 2
\end{array}
\]

\[
\begin{array}{l|llll}
  j & 1 & 2 & 3 & 4 & 5 \\
\hline
x & \text{una habitación} & \text{con} & \text{dos} & \text{camas}
\end{array}
\]

Model 3

\[
\Pr(x \mid y) = \sum_a \Pr(x, a \mid y) = \sum_a \Pr(\phi, \tau, \pi \mid y)
\]

The probability for a tablet \( \tau \) and a permutation \( \pi \) is:

\[
\Pr(\phi, \tau, \pi \mid y) = \prod_{i=1}^{I} \Pr(\phi_i \mid \phi_i^{-1}, y) \Pr(\phi_0 \mid \phi_0^{f}, y) \times \prod_{i=0}^{I} \prod_{k=1}^{I} \Pr(\tau_{i,k} \mid \tau_i^{-1}, \tau_0^{f}, \phi_0, y) \times \prod_{i=1}^{I} \prod_{k=1}^{I} \Pr(\pi_{i,k} \mid \pi_i^{-1}, \pi_0^{f}, \phi_0, y) \times \prod_{k=1}^{I} \Pr(\pi_{0,k} \mid \pi_0^{-1}, \pi_0^{f}, \phi_0, y)
\]

- \( \Pr(\phi_i \mid \phi_i^{-1}, y) \approx f(\phi_i \mid y_i) \) \text{ fertility probability}
- \( \Pr(\tau_{i,k} = x \mid \tau_i^{-1}, \tau_0^{f}, \phi_0, y) \approx l(x \mid y_i) \) \text{ lexicon probability}
- \( \Pr(\pi_{i,k} = j \mid \pi_i^{-1}, \pi_0^{f}, \phi_0, y) \approx d(j \mid i, J, I) \) \text{ distortion probability}
Model 3

- Pr(φi | φ1i−1, y) ≈ f(φi | y)
  fertility probability
- Pr(Γi,k = x | τk−1i, i−1, φi, y) ≈ l(x | y)
  lexicon probability
- Pr(Πi,k = j | πi−1, 1i−1, τ0i, φ0, y) ≈ d(j | i, J, I)
  distortion probability

\[ P_{M3}(x | y) = \sum_{(\tau, \pi) \in F(x, a)} \sum_{a} P_{M3}(\phi, \tau, \pi | y) = \]

\[ \sum_{a_1 = 0}^{I} \cdots \sum_{a_J = 0}^{I} \left( \frac{J - \phi_0}{\phi_0} \right) \left( \frac{J - 2\phi_0}{\phi_0} \right)^{J-1} \prod_{i=1}^{I} f(\phi_i | y) \prod_{j=1}^{J} l(x_j | y_{a_j}) \cdot d(j | a_j, J, I) \]

Given a target sentence y of length J,

1. For each 1 ≤ i ≤ I choose a length φi according to f(φi | y).
2. Choose a length φ0 according to f0(φ0 | \sum_{i=0}^{I} \phi_i).
3. J = \sum_{i=0}^{I} \phi_i.
4. For each 1 ≤ i ≤ I and 1 ≤ k ≤ φi, choose a source word τi,k \in \Sigma according to l(τi,k | y).
5. For each 1 ≤ i ≤ I and 1 ≤ k ≤ φi, choose a position πi,k (1 ≤ πi,k ≤ J) in the source sentence according to d(πi,k | i, J, I).
6. If any position has been chosen then error (inconsistent model).
7. For each 1 ≤ k ≤ φ0 choose a position π0,k from the vacant positions according to a uniform distribution.
An example

Given \( y: \) a double room \((I = 3)\)

Choose \( i \) using \( f(\phi | y_i) \)

Choose \( \tau_{i,k} = x \) using \( l(x | y_i) \)

Choose \( \pi_{i,k} = j \) using \( d(j | i, I, J) \)

\\[
\begin{array}{c|ccc}
   i & 1 & 2 & 3 \\
   \hline
   \phi(y_i) = \phi & 1 & 3 & 1 \\
   \tau_{i,k} = x & \text{una} & \text{con, camas, dos} & \text{habitación} \\
   \pi_{i,k} = j & 1 & 3 & 5 & 4 & 2 \\
\end{array}
\]

Choose \( j \) using \( d(j | i, I, J) \)

Choose \( x \) using \( l(x | y_i) \)

x una habitación con dos camas

Examples of alignments

Corpus EUTRANS-I: Spanish-English

1 2 3 4 5 6 7 8 9 10
por favor, ¿ podría ver alguna habitación tranquila?

• **MODEL 1, ITERATION 5**
  could (5) I (6) see (6) a (7) quiet (9) room (8) , (3) please (2) ? (4)

• **MODEL 2, ITERATION 2**
  could (5) I (6) see (6) a (7) quiet (9) room (8) , (3) please (3) ? (10)

• **MODEL 3, ITERATION 2**
  could (5) I (5) see (6) a (7) quiet (9) room (8) , (3) please (2) ? (10)
Pattern Recognition approaches to Machine Translation
Advanced Statistical Alignment Models

Model 4

For a target word $y_i$:

- The center of $y_i$, $c(i) = \sum_k \pi_{i,k} \phi_i$

- $\Pr(\phi_i \mid \phi_{i-1}^{-1}, y) \approx f(\phi_i \mid y_i)$  
  *fertility probability*

- $\Pr(\Gamma_{i,k} = x \mid \tau_{i-1}^{k-1}, \tau_0^{i-1}, \phi_0, y) \approx l(x \mid y_i)$  
  *lexicon probability*

- $\Pr(\Pi_{i,1} = j \mid \pi_1^{i-1}, \tau_0^{i-1}, \phi_0^{i-1}, y) \approx d_1(j - c(i - 1) \mid C_Y(y_{i-1}), C_X(x_j))$  
  *distortion probability for the first position in a tablet*

- $\Pr(\Pi_{i,k} = j \mid \pi_1^{i-1}, \pi_k^{i-1}, \tau_0^{i-1}, \phi_0^{i-1}, y) \approx d_{>1}(j - \pi_{i,k-1} \mid C_X(x_j))$  
  *distortion probability for the rest of positions in a tablet*

\[
y_1 \quad \ldots \quad y_{i-1} \quad y_i \quad \ldots \quad y_I
\]
\[
\phi_1 \quad \ldots \quad \phi_{i-1} \quad \phi_i \quad \ldots \quad \phi_I
\]
\[
\{x_{1,1}, \ldots, x_{1,\phi_1}\} \quad \ldots \quad \{x_{i-1,1}, \ldots, x_{i-1,\phi_{i-1}}\} \quad \{x_{i,1}, \ldots, x_{i,\phi_i}\} \quad \ldots \quad \{x_{I,1}, \ldots, x_{I,\phi_I}\}
\]
\[
\{\pi_1 < \cdots < \pi_1^{\phi_1}\} \quad \ldots \quad \{\pi_{i-1,1} < \cdots < \pi_{i-1,\phi_{i-1}}\}
\]
\[
c(1) = \frac{\sum_{r=1}^{\phi_1} \pi_1^r}{\phi_1} \quad \ldots \quad c(i - 1) = \frac{\sum_{r=1}^{\phi_{i-1}} \pi_{i-1}^r}{\phi_{i-1}}
\]

$\pi_{i,1} = j$ according to $d_1(j - c(i - 1) \mid C_Y(y_{i-1}), C_X(x_i))$

$\pi_{i,k} = j$, for $1 < k \leq \phi_i$, according to $d_{>1}(j - \pi_{i,k-1} \mid C_X(x_i))$

$\pi_{i,1} < \cdots < \pi_{i,\phi_i}$
Model 4

- $f(\phi_i | y_i)$: fertility probability
- $l(x | y_i)$: lexicon probability
- $d_{i=1}(j - c(i - 1) | C_y(y_{i-1}), C_x(x_j))$: distortion probability for the first position in a tablet
- $d_{i>1}(j - \pi_{i,k-1} | C_x(x_j))$: distortion probability for the rest of positions in a tablet

Given a target sentence $y$ of length $I$,

1. For each $1 \leq i \leq I$ choose a length $\phi_i$ according to $f(\phi_i | y_i)$.
2. Choose a length $\phi_0$ according to $f_0(\phi_0 | \sum_{i=1}^I \phi_i)$.
3. $J = \sum_{i=0}^I \phi_i$.
4. For each $1 \leq i \leq I$ and $1 \leq k \leq \phi_i$, choose a source word $\tau_{i,k}$ according to $l(\tau_{i,k} | y_i)$.
5. For each $1 \leq i \leq I$ and $1 \leq k \leq \phi_i$, choose a position $\pi_{i,k}$
   - if $k = 1$ according to $d_1$
   - if $k > 1$ according to $d_{i>1}$ but greater than $\pi_{i,k-1}$
6. If any position has been chosen then error. *(inconsistent model)*
7. For each $1 \leq k \leq \phi_0$ choose a position $\pi_{0,k}$ from the vacant positions according to a uniform distribution.

Examples of alignments

Corpus EUTRANS-I: Spanish-English

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>por favor , he hecho una reserva a nombre de Federico Redondo .</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[2-2] I (4) have (5) made (6) reservation (7) for (9) Federico (11) Redondo (12) . (0)

[4-2] I (2) am (2) leaving (2) on (5) Thursday (6) June (9) the (0) third (7) at (10) half (14) past (13) one (11) in (4) the (11) afternoon (17) . (18)

[4-5] I (2) am (2) leaving (2) on (5) Thursday (6) June (9) the (0) third (7) at (10) half (14) past (13) one (12) in (15) the (16) afternoon (17) . (18)

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>me voy a ir el jueves tres de junio a la una y media de la tarde .</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[2-2] I (2) am (2) leaving (2) on (5) Thursday (6) June (9) the (5) third (9) at (10) half (14) past (13) one (11) in (4) the (11) afternoon (17) . (18)

[4-2] I (2) am (2) leaving (2) on (5) Thursday (6) June (9) the (0) third (7) at (10) half (14) past (13) one (11) in (15) the (16) afternoon (17) . (18)

[4-5] I (2) am (2) leaving (2) on (5) Thursday (6) June (9) the (0) third (7) at (10) half (14) past (13) one (12) in (15) the (16) afternoon (17) . (18)
Model 5

For a target word $y_i$:

- Number of vacant positions up to and including position $j$ just before $\tau_{i,k}$ is placed, $v(j, \tau_{i,1}^{i-1}, \tau_{i,1}^{k-1}) \equiv v_j$.

- $Pr(\phi_i | \phi_1^{i-1}, y) \approx f(\phi_i | y_i)$ \hspace{1cm} \text{fertility probability}

- $Pr(\Gamma_{i,k} = x | \tau_{i,1}^{k-1}, \tau_0^i, \phi_0, y) \approx l(x | y_i)$ \hspace{1cm} \text{lexicon probability}

- $Pr(\Pi_{i,1} = j | \pi_1^{i-1}, \tau_0, \phi_0, y) \approx d_1(v_j | C_X(x_j), v_{c(i-1)}, v_J - \phi_i + 1) \cdot (1 - \delta(v_j, v_{j-1}))$ \hspace{1cm} \text{distortion probability for the first position in a tablet}

- $Pr(\Pi_{i,k} = j | \pi_1^{k-1}, \tau_0^i, \tau_0^i, \phi_0, y) \approx d_{>1}(v_j - v_{\pi_i,k-1} | C_X(x_j), v_J - v_{\pi_i,k-1} - \phi_i + k) \cdot (1 - \delta(v_j, v_{j-1}))$ \hspace{1cm} \text{distortion probability for the rest of positions in a tablet}

F. Casacuberta – DSIC-ITI-UPV
24-28 January 2005

Model 6

A linear combination of Model 4 and Homogeneous hidden Markov model.

$$Pr_{M6}(x | y) = \alpha \cdot Pr_{M4}(x | y) + (1 - \alpha) \cdot Pr_{HM}(x | y)$$
The training process

The training process

- Maximum likelihood by EM estimation.
- The counts in the reestimation are multiplied by $P_{RM}(x, a \mid y)$ and are added for all possible alignment.
- No efficient method is computing these estimated counts.
- The estimated counts are approximate by:
  - Computing the (approximate) most probable alignment (Model 2)
  - Apply modifications: moves and swaps
  - Sum the estimated counts for all alignments whose probability is larger than the probability of the probable alignment times a given constant.

Conventional IBM Models Training

- Every model has a specific set of free parameters.
- For example for IBM Model 4: $\theta = \{ \{l(x\mid y)\}, \{p_{=1}(\Delta_j)\}, \{p_{>1}(\Delta_j)\}, \{p(\phi\mid x)\}, p_1\}$
- To train the model parameters $\theta$: A maximum likelihood criterium, using a parallel training corpus consisting of $S$ sentence pairs $\{(x^{(n)}, y^{(n)}) : n = 1, \ldots, N\}$:
  $$\hat{\theta} = \operatorname{argmax}_\theta \prod_{n=1}^N \sum_a p_\theta(x^{(n)}, a\mid y^{(n)})$$
- The training is carried out using the Expectation-Maximization (EM) algorithm.
The EM algorithm

Given a set of pairs \((x^n, y^n)\), for \(n = 1, \ldots, N\),

- Initialize parameters \(\theta = \{l(x|y), \ldots\}\)
- Iterate (EM-procedure)
  - In the E-step, the lexicon parameter counts for every sentence pair \((y, x)\) are calculated:
    \[
    c(x|y; y, x) = N(y, x) \cdot \sum_a Pr(a|y, x) \sum_j \delta(x, x_j)\delta(y, y_{a_j})
    \]
  - In the M-step, the lexicon parameters \(\hat{l}(s|t)\) that maximize the likelihood on the training corpus are computed:
    \[
    \hat{l}(x|y) = \frac{\sum_n c(x|y; x^{(n)}, y^{(n)})}{\sum_{n,s} c(x|y; x^{(n)}, y^{(n)})}
    \]
    Similarly, the alignment/distortion and fertility parameters can be estimated for all other alignment models.
- Compute Viterbi alignments

The output is a set of aligned sentence pairs \(V(x^n, y^n); \hat{\theta}\)

Index

1 Statistical framework to machine translation \(\triangleright\) 2
2 Fertility-based models \(\triangleright\) 7
3 The search problem \(\triangleright\) 27
4 Maximum entropy models \(\triangleright\) 42
5 Using linguistic knowledge \(\triangleright\) 56
6 Bibliography \(\triangleright\) 65
The search problem in statistical machine translation

\[ \hat{y} = \arg\max_y P_r(x \mid y) \cdot P_r(y) \]

- Search is a \textbf{NP}-Hard problem. \quad (Knight, 1999)
- Algorithmic solutions: (+ heuristics for efficient suboptimal solutions)
  - Dynamic Programming \quad (Garcia-Varea, 2003) (Tillman, 2003)
  - Stack-decoding, \textit{A}\textsuperscript{*} or Branch & Bound \quad (Ortiz, 2003)
  - Greedy strategies \quad (Germann, 2001)
  - Using finite-state transducers \quad (Kumar, 2004)

Dynamic programming approach:
\textbf{DPSearchM2} for the models 3, 4 and 5*

- Using DPSearchM2 and Viterbi alignments.
- The Viterbi alignments for models 3, 4 and 5, are based on model 2.
- Solution:
  - In the final states in DPSearchM2
  - To choose the hypothesis with the best Viterbi score for a model $M$.
  - To iterate the process.
- Computational complexity: $O(J \cdot I_{\text{max}} \cdot L \cdot |E|^2)$

*The slides on searching are modified versions of some material supplied by Ismael García-Varea.
Some stack-decoding proposals

- Candide systems from IBM [Berger et al. 96]: Multiple stacks, model 3.
- Multiple stack-decoding [Wang and Waibel 98]: Model 2.
- Algorithm $A^*$ [Ueffing et al. 01]: model 4.
- Algorithm $A^*$ [Och and Ney 03]: model 6.
- Basic stack-decoding strategy:
  - Origin of the stack decoding or $A^*$: ASR
  - Optimal solution to the search problem (Jelinek, 1976)
  - Incremental development of practical hypotheses
  - The hypothesis are stored in a priority queue (a type of 'stack')
  - Selection and expansion of the top of the stack(s).

A taxonomy of the stack-decoding algorithms

- Basic stack-decoding algorithm:
  - All the hypothesis are stored in a one stack
  - A hypothesis is selected in each iteration: the hypothesis with higher score in the stack

- Problem: hypothesis with a high number of aligned words are discarded.

- Possible solutions:
  - Use of heuristics: an estimation of the contribution to the set of the optimal score.
  - Multiple stacks.

- Taxonomy:
  - Single stack algorithms $A^*$
  - Multiple stack algorithms
Basic multiple stack decoding \textit{StackDecoding}

A hypothesis in a stack:

- A prefix of the target sentence ($y^i_j$)
- A coverage subset of source positions ($C$)
- A score ($S$).

- There is one stack for each possible subset of source positions which words has already been translated.

- The possible number of stacks can be very high.

- In each iteration, the best hypothesis from each available stack is selected to generate new extended hypothesis.

- The new target prefix is the concatenation of the target prefix of the selected hypothesis and each possible target word.

- The new source positions are selected from the complementary set of $C$ (assuming some constraints).

- The new score is computed using the new ngram and the new source positions.

- The new hypothesis is stored in the corresponding stack.

Source sentence: "the configuration program"
Basic greedy algorithm \textit{GreedySearch}

- Previous works: [Germann et al. 01] for models 3 and 4.

- Characteristics:
  - Local optimization
  - No incremental building of hypothesis
  - Dependence on the initialization
  - Approximation to the search problem.
  - Fast.
  - They can be used to refine other algorithms.

- Algorithm:
  - An initial complete hypothesis and the corresponding alignment are required.
  - The hypothesis is modified iteratively until no improvements are achieved.
  - Hillclimbing algorithm
  - Building a neighbourhood from \( \langle y, a \rangle \)

- Temporal cost: \( O(J^2 \cdot |E|^2 \cdot I^2) \)
Experiments

- **EuTRANS-I corpus:**
  - Training: 10,000 pairs
  - Test: 2,636 sentences (length $\leq 15$)

- **HANSARDS corpus:**
  - Training: 128,000 pairs
  - Test: 500 sentences of 4, 6, 8, 10, 12 words

- Translation models: IBM+HMM, $1^52^53^54^55^5$

- Language models: 3-grams + smoothing *Good Turing*

- Assessment:
  - **Word Error Rate (WER):** The minimum number of substitution, insertion and deletion operations needed to convert the word string hypothesized by the translation system into a given single reference word string.
  - **Position Independent word error Rate (PER):** Similar to WER but the order is not taken into account.

The HANSARD corpus

- **Task definition:**
  - Proceedings of the Canadian parliament. (French $\rightarrow$ English)
  - Vocabulary sizes (more than two occurrences): 58,016 (French), 42,055 (English).
  - Training set: $1,7 \times 10^6$ pairs (sentence length less than 30)
  - Test set: 73 sentences.

- **First results in** (Brown et al. 1993)
  - Models:
    - 12 training iterations (1 IBM1 + 6 IBM2 + 1 IBM3 + 4 IBM5)
    - Language model: trigrams.
    - Search: stack-decoding.
  - Results:
    - 48% of sentences were successfully translated.
The **EuTRANS-I corpus**

- **Vocabulary**: 680 Spanish words, and 513 English words.
- **Training**: 10,000 pairs (97,000/99,000 words).
- **Test**: 2,996 pairs (PP=3.3) (35,000/35,590 words).

### Experimental results

<table>
<thead>
<tr>
<th>Strategy</th>
<th>sec.</th>
<th>SerErr</th>
<th>ModErr</th>
<th>Accuracy</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPSearch-M2</td>
<td>55.7</td>
<td>5.5</td>
<td>55.2</td>
<td>39.3</td>
<td>12.7</td>
<td>10.5</td>
</tr>
<tr>
<td>DPSearch-M4</td>
<td>69.5</td>
<td>12.2</td>
<td>45.1</td>
<td>42.7</td>
<td>10.2</td>
<td>9.4</td>
</tr>
<tr>
<td>StackDecoding-M4</td>
<td>87.1</td>
<td>18.4</td>
<td>44.1</td>
<td>37.5</td>
<td>14.2</td>
<td>11.1</td>
</tr>
<tr>
<td>GreedySearch-M3</td>
<td>18.7</td>
<td>61.3</td>
<td>20.5</td>
<td>18.2</td>
<td>24.8</td>
<td>18.6</td>
</tr>
<tr>
<td>GreedySearch-M4</td>
<td>165.9</td>
<td>53.0</td>
<td>23.3</td>
<td>23.7</td>
<td>20.0</td>
<td>16.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strategy</th>
<th>seg.</th>
<th>SerErrs</th>
<th>ModErr</th>
<th>Accuracy</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPSearch-M2</td>
<td>102.9</td>
<td>2.6</td>
<td>81.2</td>
<td>16.2</td>
<td>50.5</td>
<td>46.8</td>
</tr>
<tr>
<td>StackDecoding-M4</td>
<td>163.1</td>
<td>12.0</td>
<td>78.6</td>
<td>9.4</td>
<td>54.2</td>
<td>51.3</td>
</tr>
<tr>
<td>GreedySearch-M3</td>
<td>17.0</td>
<td>15.0</td>
<td>75.0</td>
<td>10.0</td>
<td>55.9</td>
<td>51.0</td>
</tr>
</tbody>
</table>
### Comparasion results

<table>
<thead>
<tr>
<th>Search strategy</th>
<th>EUTRANS-I task</th>
<th>HANSARDS task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>PER</td>
</tr>
<tr>
<td>DPSearch-M2</td>
<td>12.7</td>
<td>10.5</td>
</tr>
<tr>
<td>DPSearch-M4</td>
<td>10.2</td>
<td>9.4</td>
</tr>
<tr>
<td>StackDecoding-M4</td>
<td>14.2</td>
<td>11.1</td>
</tr>
<tr>
<td>GreedySearch-M3</td>
<td>24.8</td>
<td>18.6</td>
</tr>
<tr>
<td>GreedySearch-M4</td>
<td>20.0</td>
<td>16.2</td>
</tr>
<tr>
<td>SWB&lt;sup&gt;(1)&lt;/sup&gt;</td>
<td>10.8</td>
<td>10.0</td>
</tr>
<tr>
<td>SWB+IBM&lt;sup&gt;(1)&lt;/sup&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT&lt;sup&gt;(1)&lt;/sup&gt;</td>
<td>4.4</td>
<td>2.9</td>
</tr>
<tr>
<td>A&lt;sup&gt;*&lt;/sup&gt;&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A&lt;sup&gt;*&lt;/sup&gt;&lt;sup&gt;(3)&lt;/sup&gt;-M4</td>
<td>5.5</td>
<td></td>
</tr>
</tbody>
</table>

<sup>(1)</sup> In [Och 02] (Ph.D.)

<sup>(2)</sup> In [Ueffing et al. 02] for sentences (≤ 12) and computational time of 127 sec.

<sup>(3)</sup> In [Prat 02].

### Results with categories

<table>
<thead>
<tr>
<th>Model</th>
<th>EUTRANS-I task</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment templates</td>
<td>manual</td>
<td>2.5</td>
</tr>
<tr>
<td>STM category translation + A&lt;sup&gt;*&lt;/sup&gt; with M4</td>
<td>automatic</td>
<td>3.8</td>
</tr>
<tr>
<td>Alignment templates</td>
<td>automatic</td>
<td>4.4</td>
</tr>
<tr>
<td>A&lt;sup&gt;*&lt;/sup&gt; with M4</td>
<td>no</td>
<td>5.3</td>
</tr>
<tr>
<td>Quasi-monotone search</td>
<td>manual</td>
<td>6.7</td>
</tr>
<tr>
<td>STM</td>
<td>no</td>
<td>7.0</td>
</tr>
<tr>
<td>DP-search with M2</td>
<td>manual</td>
<td>9.8</td>
</tr>
<tr>
<td>Quasi-monotone search</td>
<td>no</td>
<td>10.8</td>
</tr>
<tr>
<td>DP-search with M2</td>
<td>no</td>
<td>13.9</td>
</tr>
</tbody>
</table>
The performance of a statistical machine translation system depends on the quality of lexicon and alignment models used.

Typically, these statistical alignment models are based on single-word dependencies → lack of useful context information that can lead to inadequate alignments.

A possible solution would be to include more dependencies in the lexicon model i.e. \( l(x_j|y_{a_{j-1}}, y_{a_j}) \Rightarrow \) problem: significant data sparseness.

A possible solution: Use maximum entropy to build context-dependent lexicon models.

Some advantages of using maximum entropy
- Easy to integrate additional knowledge sources
- No problem with overlapping features
- Well-founded mathematical theory
- Efficient training algorithms
- ...
Maximum entropy principle

- A model that takes a context $w$ into account $\Rightarrow p_y(x|w)$ instead of $l(x|y)$.

- The properties that can be useful: by feature functions $\phi_{y,k}(w, x), k = 1, \ldots, K_y$.
  - For example, to model the existence or absence of a specific target word $y'$ in the context of a target word $y$, which can be translated by the source word $x'$.
  - This dependence using the following indicator function (feature):
    \[
    \phi_{y,1}(w, x) = \begin{cases} 
    1 & \text{if } x = x' \text{ and } y' \in w \\
    0 & \text{otherwise}
    \end{cases}
    \]

Consequently the first feature for word $y$ has associated the pair $(y', x')$.

Maximum entropy principle

- The entropy maximum principle suggests that the optimal parametric form of a model $p_y(x|w)$ taking into account the feature functions $\phi_{y,k}, k = 1, \ldots, K_y$ is given by:
  \[
  p_y(x|w) = \frac{1}{Z_{\Lambda_y}(w)} \cdot \exp \left\{ \sum_{k=1}^{K_y} \lambda_{y,k} \cdot \phi_{y,k}(w, x) \right\}
  \]

- The resulting model has an exponential form with free parameters:
  \[
  \Lambda_y \equiv \{ \lambda_{y,k}, k = 1, \ldots, K_y \}
  \]

- The parameter values that maximize the likelihood for a given training corpus can be computed using the so-called GIS algorithm.
Contextual information and features definition

- A model $p_y(x|w)$ and a sample training for each target word $y$ are needed.
- In a pair of sentences $(x, y)$, contextual information (easily extended):
  - Target context: $y_{i-3}...y_i...y_{i+3}$
  - Source context: $x_j$
  - Word classes: syntactic and semantic information $(T(y_i), S(x_j))$.
- Feature categories:

<table>
<thead>
<tr>
<th>Category</th>
<th>$\phi_{y_i,k}(w, x) = 1$ if and only if ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x_j = \Diamond$ and $\Box \in y_i$</td>
</tr>
<tr>
<td>2</td>
<td>$x_j = \Diamond$ and $\Box \in y_i$</td>
</tr>
<tr>
<td>3</td>
<td>$x_j = \Diamond$ and $\Box \in y_i$</td>
</tr>
<tr>
<td>4</td>
<td>$x_j = \Diamond$ and $\Box \in y_i$</td>
</tr>
<tr>
<td>5</td>
<td>$x_j = \Diamond$ and $\Box \in y_i$</td>
</tr>
</tbody>
</table>

In all cases the $k$-th feature has associated the pair $(\Diamond, \Box)$.

Maximum entropy models training integration

- Model parameters to be learnt: $\Lambda_t \equiv \{\lambda_{t,k} : k = 1, \ldots, K\}$
- In the E-step, a refined count collection for the lexicon parameters is performed
  $$c(x|y, w; x, y) = N(x, y) \cdot \sum_a Pr(a|x, y) \sum_j \delta(x, x_j)\delta(y, y_{a_j})\delta(w, w_{j, a_j})$$
  $w_{j,a_j}$ $\equiv$ the maximum entropy context that surrounds $x_j$ and $y_{a_j}$
- In the M-step, the new lexicon parameters are computed:
  $$\hat{\Lambda}_y = \arg\max_{\Lambda_y} \prod_{x, w} c(s|y, w; x, y) \cdot \log p_y(x|w)$$
  $c(x|y, w; x, y) \equiv$ weights of the training samples $(x, y, w)$ used to train the maximum entropy model (number of times that $(x, y, w)$ occurs).
- The re-estimation of the alignment/distortion and fertility probabilities does not change if we use a maximum entropy lexicon model.
The EM-ME algorithm

Given a set of pairs \((x^n, y^n)\), for \(n = 1, \ldots, N\),

- Initialize parameters \(\theta = \{l(x|y), \ldots\}\)

- Iterate (EM-procedure)
  - In the E-step:
    1. Collect counts for alignment/distortion and fertility parameters.
    2. Collect refined lexicon counts (Overhead on space and computation time).
  - In the M-step:
    1. Reestimate alignment/distortion and fertility parameters.
    2. Perform GIS training for lexicon parameters (Overhead on space and computation time).

- Compute Viterbi alignments

  The output is a set of aligned sentence pairs \(V(x^n, y^n); \hat{\theta}; \hat{\Lambda}_y\)

Potential problems of the ME-EM integration

- Computation overhead:
  - In the \(k\)-th iteration of the E-step
  - In the M-step the computation of the GIS training for each word

- Space overhead:
  We have to store every possible maximum entropy training event \((s, t, x)\), that is, every possible combination of \(t \in V_T\), \(s \in V_S\) and \(x \Rightarrow\) requires a huge quantity of memory.
Experimental results

- Efficiency: time consumption of different approaches.
- Performance: comparison of the alignment quality (of 500 randomly selected pairs) obtained with all the IBM models (1 to 5) with and without using maximum entropy modeling.

- Tasks: Verbmobil and Hansards

<table>
<thead>
<tr>
<th></th>
<th>Verbmobil</th>
<th></th>
<th>Hansards</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>German</td>
<td>English</td>
<td>French</td>
<td>English</td>
</tr>
<tr>
<td>Training Sentences</td>
<td>34,446</td>
<td></td>
<td>1,470K</td>
<td></td>
</tr>
<tr>
<td>Words</td>
<td>329,625</td>
<td>343,076</td>
<td>24.33M</td>
<td>22.16M</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>5,936</td>
<td>3,505</td>
<td>100,269</td>
<td>78,332</td>
</tr>
</tbody>
</table>

Time consumption results

- Time consumption in seconds of different approaches per EM iteration (on average for the five IBM models) for different sizes of training corpus.

<table>
<thead>
<tr>
<th>Task</th>
<th>Size of train</th>
<th># of ( \ell )</th>
<th>Conventional train</th>
<th>ME-train</th>
<th>Simplified ME-train</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verbmobil</td>
<td>0.5K</td>
<td>29</td>
<td>1</td>
<td>29</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>8K</td>
<td>84</td>
<td>18</td>
<td>235</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>35K</td>
<td>209</td>
<td>60</td>
<td>2290</td>
<td>675</td>
</tr>
<tr>
<td>Hansards</td>
<td>0.5K</td>
<td>15</td>
<td>2.5</td>
<td>29</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8K</td>
<td>80</td>
<td>35</td>
<td>1180</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>128K</td>
<td>1214</td>
<td>655</td>
<td>16890</td>
<td>6870</td>
</tr>
</tbody>
</table>
Alignment quality results: evaluation methodology

- An annotation scheme that explicitly allows for ambiguous alignments.
- Two different kinds of alignments: a $S(ure)$ alignment (unambiguous ■) and a $P(ossible)$ alignment (ambiguous □).
- The $P$ labels are used specially to align words within idiomatic expressions, free translations, and missing function words ($S \subseteq P$).
- Reference alignment: many-to-one and one-to-many relationships.

Example of a manual alignment

The quality of an alignment $A = \{(j, a_j) | a_j > 0\}$ is then computed by appropriately redefined precision and recall measures:

$$\text{recall} = \frac{|A \cap S|}{|S|}, \quad \text{precision} = \frac{|A \cap P|}{|A|}$$

and using the following alignment error rate, which is derived from the well known F-measure:

$$AER(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

- In such a way, a recall error can only occur if a $S(ure)$ alignment is not found and a precision error can only occur if the found alignment is not even $P(ossible)$. 

Alignment quality results: AER

- AER of 500 randomly selected sentence pairs.

<table>
<thead>
<tr>
<th>Training Scheme</th>
<th>Model</th>
<th>Hansards task</th>
<th>Verbmbil task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Size of train corpus</td>
<td>Size of train corpus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5K</td>
<td>8K</td>
</tr>
<tr>
<td>1^5</td>
<td>1</td>
<td>48.0</td>
<td>35.1</td>
</tr>
<tr>
<td></td>
<td>1+ME</td>
<td>47.7</td>
<td>32.7</td>
</tr>
<tr>
<td>1^52^5</td>
<td>2</td>
<td>46.0</td>
<td>29.2</td>
</tr>
<tr>
<td></td>
<td>2+ME</td>
<td>44.7</td>
<td>28.0</td>
</tr>
<tr>
<td>1^52^53^3</td>
<td>3</td>
<td>43.2</td>
<td>27.3</td>
</tr>
<tr>
<td></td>
<td>3+ME</td>
<td>42.5</td>
<td>26.4</td>
</tr>
<tr>
<td>1^52^53^34^3</td>
<td>4</td>
<td>41.8</td>
<td>24.9</td>
</tr>
<tr>
<td></td>
<td>4+ME</td>
<td>41.5</td>
<td>24.3</td>
</tr>
<tr>
<td>1^52^53^34^35^3</td>
<td>5</td>
<td>41.5</td>
<td>24.8</td>
</tr>
<tr>
<td></td>
<td>5+ME</td>
<td>41.5</td>
<td>24.5</td>
</tr>
</tbody>
</table>

- The alignment error rate improves using the context-dependent lexicon models.

- For the Verbmobil task, the improvements are smaller than for the Hansards task, which might be due to the fact that already the baseline alignment quality is very good.

Precision and recall [%] results for Hansards task for different corpus sizes in every iteration of the training:

![Precision vs. Training scheme](image1)
![Recall vs. Training scheme](image2)
Is the linguistic knowledge needed for statistical machine translation?

- **YES?**
  - There are many linguistic knowledge available.
  - The bilingual training data can be better exploited.

- **NOT?**
  - Many linguistic knowledge is hard to formalize.
  - The generation of new linguistic knowledge requires great human effort.
Linguistic knowledge that has been used in statistical machine translation

- Morpho-syntactic knowledge: lexicon, Part-of-Speech, etc... (Nießen and Ney, 2004)

  Hybrid linguistic-statistical approaches have been used with success (i.e. hidden markov models)

- Others: Cognates (Kondrak, Marcu and Knight, 2003), named entities (Huang, Vogel and Waibel, 2003), ...

- Syntactic information: next talk!

Morpho-syntactic knowledge in statistical machine translation


- Present statistical machine translation systems often treat different inflected forms of the same lemma as if they were independent of one another.

- The bilingual data can be better exploited by explicitly taking into account the interdependencies of related inflected forms.

A possible proposal: HIERARCHICAL LEXICON MODELS
Morpho-syntactic knowledge in statistical machine translation

\[ yo \text{ como} \text{ pan} \]

- Morphological and syntactic tags (POS, tense, person, ...)
- The base form

\[ T = t^6_1 = \text{comer} \text{ verb indicative present singular 1} \]

\[
\Pr(x | y) = \sum_{a \in A(y, x)} \Pr(J | y) \cdot \Pr(a | J, y) \cdot \Pr(x | a, J, y)
\]

\[
(t^n_1)_j \equiv T_j
\]

\[
\Pr(x | a, J, y) = \sum_{T^J_1} \Pr(x, T^J_1 | a, J, y)
\]

\[= \sum_{T^J_1} \prod_{j=1}^{J} \Pr(x_j, T_j | x^j_1, T^J_1, a, J, y)\]

\[\approx \sum_{T^J_1} \prod_{j=1}^{J} l(x_j, T_j | y_a_j)\]

A lemma-tag lexicon: \(l(x_j, T_j | y_a_j)\)
Estimation of the lemma-tag lexicon

Maximum entropy modelling

\[ l(s, T \mid t) \equiv l_\Lambda(s, T \mid t) = \frac{\exp \left[ \sum_m \lambda_m h_m(t, s, t^n) \right]}{\sum_{\tilde{s}, \tilde{t}^n} \exp \left[ \sum_m \lambda_m h_m(t, \tilde{s}, \tilde{t}^n) \right]} \]

\[ \Lambda = \{ \lambda_m \} \]

- During training, the sum on $\tilde{s}$ and $\tilde{t}^n$ is restricted to the reading of word forms having the same base form and partial reading as a word forms aligned at least once with $t$.


Experiments


- Verbmobil task:
  - Automatic translation of spontaneously spoken dialogs (English $\rightarrow$ German)
  - Vocabulary sizes: 1. 4,674 word forms (English) and 7,940 word forms (German).  
    2. 3,639 base forms (English) and 6,063 base forms (German)
  - Training set: 58,073 pairs (549Kw/519Kw).
  - Test set: 527 sentences.

- Results (m-WER):
  - 31.8% (34.1% in the baseline).
Open problems

• Automatically induction of the morphology of inflectional languages using only text corpora and no human input: Using prefix trees (Schone and Jurafsky, 2000) or pairs of hidden Markov models (Clark, 2000)

• Using “conventional” dictionaries (collections of word or phrase pairs collected by hand) (Nießen and Ney, 2004)

• Unknown words by some semantic information of the context words (Widdows, 2003).

• Extracting named entity translingual equivalences from bilingual parallel corpora (Huang, Vogel and Waibel, 2003)

• Using cognates (Kondrak, Marcu and Knight, 2003)

Index

1 Statistical framework to machine translation ▶ 2

2 Fertility-based models ▶ 7

3 The search problem ▶ 27

4 Maximum entropy models ▶ 42

5 Using linguistic knowledge ▶ 56

6 Bibliography ▶ 65
Bibliography


Pattern Recognition approaches to Machine Translation

F. Casacuberta and E. Vidal

Pattern Recognition and Human Language Technology Group
Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain

Stochastic Finite-State Translation Models

Enrique Vidal
evital@iti.upv.es

January 2005

E. Vidal – ITI-UPV-DSIC

Index

1 Introduction 2
2 Rational or Finite-State Transduction 6
3 Stochastic Finite-State Transducers 11
4 Error Correcting 20
5 Sequential Transduction 26
6 Subsequential Transduction:
   Introduction to the “OSTI” Algorithm 32
7 Bibliography 36
Pattern Recognition, Natural Language Processing
and Finite-State Transduction

- (Stochastic) Grammars and Automata are adequate models for Classification tasks. But there are many Pattern Recognition problems which are better framed within the most general Interpretation paradigm
- Interpretation tasks and be conceptually (and practically) tackled through Formal Transduction.
- E.g., many Continuous Speech Recognition and Understanding tasks can be seen as (simple) transductions from certain acoustic, phonetic or lexical input sequences into output sequences of higher-level linguistic categories
- Many direct applications such as Language Translation and Semantic Decoding
- Simple transducers are often powerful enough to deal with useful mappings between complex languages
Probabilistic problem statement

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

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

The joint probability $\Pr(x, y)$ can be adequately modelled by means of a stochastic finite-state transducer $T$:

$$
\Pr(x, y) \approx P_T(x, y)
$$

However, not all the transduction tasks are equally difficult . . .

Not all the transduction tasks are equally difficult: examples

- 1... Spanish to English, word by word
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?

- 2... Division by 7
  3 5 7 6 8 1 8 0 3 1 ( : 7 = )

- 3... English to Decimal
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE

- 4... Roman to Decimal
  III XIX XLII LXXIV CDII CMLXXXIX

- 5... ATIS: English to "Pseudo English"
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- 6... Spanish to English
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

- 1... Spanish to English, word by word
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to

- 2... Division by 7
  3 5 7 6 8 1 8 0 3 1 (: 7 = )

- 3... English to Decimal
  NINEHUNDREDANDNINETEENHUNDREDANDNINE

- 4... Roman to Decimal
  III XIX XLII LXXIV CDII CMLXXXIX

- 5... ATIS: English to ”Pseudo English”
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- 6... Spanish to English
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

• 1... Spanish to English, word by word
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
   to what time

• 2... Division by 7
   3 5 7 6 8 1 8 0 3 1 ( : 7 = )

• 3... English to Decimal
   NINEHUNDREDANDNINETEENTHOUSANDANDNINE

• 4... Roman to Decimal
   III XIX XLII LXXIV CDII CMLXXXIX

• 5... ATIS: English to "Pseudo English"
   WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

• 6... Spanish to English
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

- 1... Spanish to English, word by word
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to what time departs the flight more early of Boston to Denver in TWA?

- 2... Division by 7
  \[ \begin{array}{cccccc}
  3 & 5 & 7 & 6 & 8 & 1 \\
  \end{array} \]
  \( : 7 = \)
  \[ \begin{array}{cccccc}
  0 & 5 & 1 & 0 & 9 & 7 \\
  \end{array} \]

- 3... English to Decimal
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE

- 4... Roman to Decimal
  \[ \begin{array}{cccccc}
  III & XIX & XLII & LXXIV & CDII & CMLXXXIX \\
  \end{array} \]

- 5... ATIS: English to "Pseudo English"
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- 6... Spanish to English
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

- 1... Spanish to English, word by word
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to what time departs the flight more early of Boston to Denver in TWA?

- 2... Division by 7
  3 5 7 6 8 1 8 0 3 1 ($: 7 = $)
  0 5

- 3... English to Decimal
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE

- 4... Roman to Decimal
  III XIX XLII LXXIV CDII CMLXXXIX

- 5... ATIS: English to "Pseudo English"
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- 6... Spanish to English
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

1. Spanish to English, word by word
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
   to what time departs the flight more early of Boston to Denver in TWA?

2. Division by 7
   3 5 7 6 8 1 8 0 3 1 ( : 7 = )
   0 5 1 0

3. English to Decimal
   NINEHUNDREDANDNINETEENTHOUSANDANDNINE

4. Roman to Decimal
   III XIX XLII LXXIV CDII CMLXXXIX

5. ATIS: English to "Pseudo English"
   WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

6. Spanish to English
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

- 1... Spanish to English, word by word
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to what time departs the flight more early of Boston to Denver in TWA?

- 2... Division by 7
  \[
  \begin{array}{l}
  3 & 5 & 7 & 6 & 8 & 1 & 8 & 0 & 3 & 1 \\
  0 & 5 & 1 & 0 & 9 & 7 & 4 & 0 & 0 & 4
  \end{array}
  \quad (: 7 = )
  \]

- 3... English to Decimal
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE

- 4... Roman to Decimal
  \[
  \begin{array}{llllll}
  \text{III} & \text{XIX} & \text{XLII} & \text{LXXIV} & \text{CDII} & \text{CMLXXXIX} \\
  \end{array}
  \]

- 5... ATIS: English to "Pseudo English"
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- 6... Spanish to English
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?

E. Vidal – ITI-UPV-DSIC January 2005 Page 4.4
Not all the transduction tasks are equally difficult: examples

- 1... Spanish to English, word by word
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to what time departs the flight more early of Boston to Denver in TWA?

- 2... Division by 7
  \[3 5 7 6 8 1 8 0 3 1 (\div 7 =)\]
  \[0 5 1 0 9 7 4 0 0 4\]

- 3... English to Decimal
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE

- 4... Roman to Decimal
  \[III \ XIX \ XLII \ LXXIV \ CDII \ CMLXXXIX\]

- 5... ATIS: English to "Pseudo English"
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- 6... Spanish to English
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

1. Spanish to English, word by word
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
   to what time departs the flight more early of Boston to Denver in TWA?

2. Division by 7
   \[\begin{array}{cccccccc}
   3 & 5 & 7 & 6 & 8 & 1 & 8 & 0 & 3 & 1 \\
   0 & 5 & 1 & 0 & 9 & 7 & 4 & 0 & 0 & 4 \\
   \end{array}\] (\(\div 7 =\))

3. English to Decimal
   NINEHUNDREDANDNINETEENTHOUSANDANDNINE
   9

4. Roman to Decimal
   III XIX XLII LXXIV CDII CMLXXXIX

5. ATIS: English to "Pseudo English"
   WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

6. Spanish to English
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

1. Spanish to English, word by word
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
   to what time departs the flight more early of Boston to Denver in TWA?

2. Division by 7
   3 5 7 6 8 1 8 0 3 1 ( : 7 = )
   0 5 1 0 9 7 4 0 0 4

3. English to Decimal
   NINEHUNDREDANDNINETEENTHOUANDANDNINE
   9 19

4. Roman to Decimal
   III XIX XLII LXXIV CDII CMLXXXIX

5. ATIS: English to "Pseudo English"
   WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

6. Spanish to English
   ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

- **1... Spanish to English, word by word**
  
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  
  to what time departs the flight more early of Boston to Denver in TWA?

- **2... Division by 7**
  
  \[
  \begin{array}{cccccccccc}
  3 & 5 & 7 & 6 & 8 & 1 & 8 & 0 & 3 & 1 \\
  0 & 5 & 1 & 0 & 9 & 7 & 4 & 0 & 0 & 4 \\
  \end{array}
  \]
  
  \( : 7 = \)

- **3... English to Decimal**
  
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE
  
  9 19 0 0 4

- **4... Roman to Decimal**
  
  III XIX XLII LXXIV CDII CMLXXXIX

- **5... ATIS: English to "Pseudo English"**

  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- **6... Spanish to English**
  
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

1... Spanish to English, word by word
¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
to what time departs the flight more early of Boston to Denver in TWA?

2... Division by 7
3 5 7 6 8 1 8 0 3 1 ( : 7 = )
0 5 1 0 9 7 4 0 0 4

3... English to Decimal
NINEHUNDREDANDNINETEENOUSANDANDNINE
9 19 0

4... Roman to Decimal
III XIX XLII LXXIV CDII CMLXXXIX

5... ATIS: English to "Pseudo English"
WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

6... Spanish to English
¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

- 1... Spanish to English, word by word
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to what time departs the flight more early of Boston to Denver in TWA?

- 2... Division by 7
  \[ \begin{array}{cccccccc}
  3 & 5 & 7 & 6 & 8 & 1 & 8 & 0 & 3 \ 1 \\
  \end{array} \quad ( : 7 = ) \begin{array}{cccccccc}
  0 & 5 & 1 & 0 & 9 & 7 & 4 & 0 & 0 \ 4 \\
  \end{array} \]

- 3... English to Decimal
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE
  9 \quad 19 \quad 0 \quad 09

- 4... Roman to Decimal
  III \quad XIX \quad XLII \quad LXXIV \quad CDII \quad CMLXXXIX

- 5... ATIS: English to "Pseudo English"
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?

- 6... Spanish to English
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
Not all the transduction tasks are equally difficult: examples

- **1... Spanish to English, word by word**
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to what time departs the flight more early of Boston to Denver in TWA?

- **2... Division by 7**
  \[
  \begin{array}{cccccccccc}
  3 & 5 & 7 & 6 & 8 & 1 & 8 & 0 & 3 & 1 \\
  0 & 5 & 1 & 0 & 9 & 7 & 4 & 0 & 0 & 4 \\
  \end{array}
  \quad (: 7 = )
  \]

- **3... English to Decimal**
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE
  \[
  \begin{array}{cccc}
  9 & 19 & 0 & 09 \\
  \end{array}
  \]

- **4... Roman to Decimal**
  III XIX XLII LXXIV CDII CMLXXXIX
  \[
  \begin{array}{cccccccc}
  3 & 19 & 4 & 2 & 74 & 4 & 02 & 9 & 8 & 9 \\
  \end{array}
  \]

- **5... ATIS: English to "Pseudo English"**
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?
  List departure time of earliest morning TWA flights from Boston and to Denver

- **6... Spanish to English**
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  What is the departure time of TWA earliest flight from Boston to Denver?
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.

Index

1. Introduction  2
2. Rational or Finite-State Transduction  6
3. Stochastic Finite-State Transducers  11
4. Error Correcting  20
5. Sequential Transduction  26
6. Subsequential Transduction:
   Introduction to the “OSTI” Algorithm  32
7. Bibliography  36
Finite State Transducers (FST): formal definition

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
- $Q_F \subseteq Q$: Set of Final States
- $E \subseteq Q \times X^* \times Y^* \times Q$: "Edges" or Transitions

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

**EXAMPLE**

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

Three possible types of ambiguity: input, output and path

Another example of a (very small) FST for a toy, but real task

(Learned from MLA Spanish-English training sentences with OSTIADR using input and output 4-Gram constraints)
Finite State Transducers: Paths and Translations

- A path $P$ of a Finite State transducer $\tau = (Q, X, Y, q_0, Q_F, E)$ is a sequence of transitions of $E$.

- A translation of $\tau$ is a pair of strings $(x, y) \in X^* \times Y^*$ such that there is a path $P$ in $\tau$ which “matches” $x$ and $y$; that is:

$$P = (q'_1, u_1, v_1, q_1), (q'_2, u_2, v_2, q_2), \ldots, (q'_m, u_m, v_m, q_m)$$

where $q'_1 = q_0$, $q_i = q'_{i+1}$ for $1 \leq i < m$, $q_m \in Q_F$, $x = u_1 \cdots u_m$, $y = v_1 \cdots v_m$.

- $T_\tau \subset X^* \times Y^*$: $T_\tau = \{(x, y) \in X^* \times Y^* : (x, y) \text{ are translations of } \tau\}$

- Let $P(\tau, x, y)$ be the set of matching paths of $x, y$ in $\tau$.

$\tau$ is ambiguous if $\exists x', y'$ such that $|P(\tau, x', y')| > 1$.

Example: $P(\tau, bcc, 1113) = \{(A, b, 1, C)(C, c, 1, C)(C, \lambda, 3, B), (A, b, 1, c)(C, c, 11, E)(C, \lambda, 3, B)\}$

Finite State Transducer Learning and Grammatical Inference

- A Finite State (regular) Grammar (FSG), $G$, can be seen as a particular case of Finite State Transducer (FST), $T$ which, for each input string $x$, produces an output string $y$, such that $y = \text{YES}$ if $x$ belongs to the language of $G$ and $y = \text{NO}$ otherwise.

- Any algorithm that would learn any FST could also learn any FSG and, therefore, learning Finite State Transducers (FST) is at least as hard as learning Finite State (regular) Grammars (FSG).

- Transducer Learning can be properly framed within the paradigm of Grammatical Inference.

Transducer Identification in the Limit:

Let $f : X^* \rightarrow Y^*$ be a transduction function. A transducer learning algorithm $\mathcal{A}$ is said to identify $f$ in the limit if, for any positive presentation $S$ of input-output pairs of $f$, $\mathcal{A}$ converges to a transduction $g : X^* \rightarrow Y^*$ such that $\forall x \in \text{Dom}(f), g(x) = f(x)$, when the number of pairs in $S$ tends to infinity.
Stochastic Finite State Transducers

A Stochastic Finite State transducer $T$ is defined by $(\tau, P, P_F)$, where:

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

- Probability of a path, $P_m$, ending at the state $q_m$:
  $$Pr(P_m) = \prod_{(q', u, v, q) \in P_m} P(q', u, v, q) P_F(q_m)$$

- Probability of a translation $(x, y)$ of $\tau$:
  $$P_T(x, y) = \sum_{P_m \in \mathcal{P}(x, y)} Pr(P_m) = \sum_{P_m \in \mathcal{P}(x, y)} \prod_{(q', u, v, q) \in P_m} P(q', u, v, q) P_F(q_m)$$

$P_T(x, y)$ defines a joint distribution in $X^*, Y^*$
Example of a Stochastic Finite-State Transducer

\[
Pr(\text{una camera doppia, a double room}) = 0.5 \cdot 0.6 \cdot 0.3 = 0.09
\]

\[
Pr(\text{una camera doppia, a room with two beds}) = 0.5 \cdot 0.1 \cdot 1.0 = 0.05
\]

Stochastic Finite-State Transducer: another example

\[
Pr(\text{una camera singola, a single room}) = 0.5 \cdot 0.1 \cdot 0.2 + 0.5 \cdot 0.6 \cdot 0.7 = 0.01 + 0.21 = 0.22
\]
Stochastic Finite State Transducers: embeeded language models

The marginals of the joint probability distribution $P_T(x, y)$ defined by a stochastic finite-state transducer $T$ are stochastic regular languages:

$$P_i(x) = \sum_{y \in Y^*} P_T(x, y),$$
$$P_o(y) = \sum_{x \in X^*} P_T(x, y).$$

These languages can be properly considered as input and output Language Models corresponding to $T$.

In practice, these Language Models are simply the regular languages associated to the automata obtained by dropping the input and output symbols of each transition of the finite-state transducer, respectively.

Stochastic Finite State Transducers: search problems

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

  $$\hat{P} = \arg\max_{P \in \mathcal{P}(\tau, x', y') \; ; \; x' = x, y' = y} Pr(P)$$

  *Efficient solution by Dynamic Programming*

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

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

  *No efficient solution* (shown to be NP-Hard!).

  *Approximation:*

  $$\tilde{y} = \arg\max_{P \in \mathcal{P}(\tau, x', y') \; ; \; x' = x, y' \in Y^*} Pr(P)$$

  *Efficient solution by Viterbi search*

Both problems are easy if $\tau$ is un-ambiguous – trivial if $\tau$ is deterministic
### Example of Viterbi translation

```
una camera doppia
```

\[
\arg\max_y \Pr(\text{"una camera doppia"}, y) \approx \text{"a double room"}
\]

---

### 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:**
  - **Back-off and interpolation**
    Adapted from techniques used in language modeling [Llorens 01] (so far fully developed only for techniques based on N-Grams)
  - **Stochastic error-correcting parsing**
    Given a source sentence, $x$, find a path in the transducer that error-correcting matches $x$ with maximum probability
Error Correcting techniques: Motivation

Often needed to allow parsing unseen input sentences through learned Finite State models that do not completely "cover" the input language:

- Can be understood as a kind of smoothing that can be applied to most types of Finite State devices.
- Explicitly copes with "imperfect" input sentences (i.e., sentences inappropriately modeled by the trained models).
- Also copes with insufficiently trained models. In an extreme view, this is similar to Memory Based techniques, where only the (raw) training data is considered (no generalization).

Finite State Error Correcting Parsing

- Each input sentence, $x$, is considered as a corrupted version of some sentence $x' \in L$, where $L$ is the language (domain) of the FSM.
- The corruption process is modelled by means of an Error Model $E$, that accounts for (single-word) substitutions, insertions and deletions.
- The parsing of $x$ consists in finding a string $\hat{x}$ in $L$ which has maximum posterior probability of having been distorted into $x$; that is,

$$\hat{x} = \arg\max_{x' \in L} P(x'|x) = \arg\max_{x' \in L} P_L(x') \cdot P_E(x|x')$$

where $P_L(x')$ is the probability of $x'$ in $L$, given by the (input part of the) FSM, and $P_E(x|x')$ is the probability of $x$ being a corrupted version of $x'$ according to $E$.

The resulting translation, $y'$, is the string associated to $\hat{x}$ through the SST.
Finite State Error Correcting Parsing: training and search

- The parameters of $P_E(x|x')$ are estimated from a set of “distorted” sentences; i.e., sentences that cannot be exactly parsed through the given FST.

- If both $P_E(x|x')$ and $P_L(x)$ are given by a Finite-State models, the Error Correcting search

$$\hat{x} = \arg\max_{x' \in L} P_L(x') \cdot P_E(x|x')$$

can be efficiently performed through appropriate extensions of the Viterbi Algorithm [Amengual & Vidal, PAMI-1999]

Efficient Finite State Error Correcting Parsing

The required search can be efficiently performed through appropriate extensions of the Viterbi Algorithm:

- For every input language word, a loop transition is added to each state of the FSM to account for insertion-errors.

- Each transition is expanded with the appropriate substitution-error transitions plus an empty transition to account for a deletion-error.

- The standard Viterbi trellis is extended with “horizontal” arcs for the insertion-errors and “vertical” arcs for the deletion-errors.

- Actual expansion of the trellis is not necessary (nor generally possible!). Virtual expansion is one of the key issues for efficiency.

- Efficient techniques can be used to process the scores in each stage of the trellis, overcoming the trouble raised by vertical arcs [Amengual & Vidal, PAMI-1999].
Error Correcting Parsing: example

Grammar, equivalent Automaton and (virtual) Error-Correcting extensions

Ins-Sub-Del-Extended trellies and Error-Correcting Parsing process

Index

1 Introduction

2 Rational or Finite-State Transduction

3 Stochastic Finite-State Transducers

4 Error Correcting

5 Sequential Transduction

6 Subsequential Transduction:
   Introduction to the “OSTI” Algorithm

7 Bibliography
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.

Sequential Transducers

A Sequential Transducer (ST) τ is a 5-tuple \( \tau = (Q, X, Y, q_0, E) \):

- \( Q \): Finite set of States
- \( X, Y \): Input and output Alphabets
- \( q_0 \in Q \): Initial State
- \( E \subseteq Q \times X \times Y^* \times Q \): "Edges" or Transitions

- All the states are accepting
- Edges are deterministic:
  \[ (q, a, u, r), (q, a, v, s) \in E \Rightarrow (u = v \land r = s) \]

**Properties:**

1. \( T_\tau \) is a function: \( X^* \rightarrow Y^* \)
2. STs \( \equiv \) Generalized Sequential Machines \( \supseteq \) (Mealy and Moore machines)
3. STs preserve prefixes: \( T_\tau(\lambda) = \lambda; \ T_\tau(uv) \in T_\tau(u)Y^* \)

"Property" 2 entails strict sequentiality, which can hardly be adequate in many cases of interest.
**An example of Sequential Transduction; sequential segmentation**

\[ X = \{a, b\}; \quad Y = \{A, B\} \]

\[ \tau = \frac{a}{\lambda} \quad \frac{b}{\lambda} \quad \frac{b}{\lambda} \quad \frac{a}{A} \]

\[ T_{\tau} = \{(\lambda, \lambda), (aba, A), (abababa, ABA), (ababababb, ABAB), \ldots\} \]

Sequential segmentation of the input string “abababababa”

```
| a | b | a | b | A | b | a | b | A | a | b | A |
```

**Learning Sequential Transducers using language learning (GI) techniques**
Applications of Sequential Transduction: Language Understanding using “intermediate” semantic languages

Basic idea: Split the single-block transducer $T$ into two blocks, using an Intermediate Semantic Language (ISL) which is sequential with the input.

**Example (Spanish numbers to Decimal)**

- **Input:** doscientos doce mil dieciseseis
- **ISL:** $+2 \times 100 + 2 + 10 ) \times 1000 ( + 10 + 6$

**Example (From the ATIS task)**

- **Input:** I’d like to fly from Boston to Denver with American Airlines on Tuesday
- **ISL:** $\text{REQ} = \text{FLIGHTS} \quad \text{ORG} = \text{BBOS} \quad \text{DST} = \text{DDEN} \quad \text{AIRLINE} = \text{AA} \quad \text{WEEKDAY} = \text{TU}$

---

**Index**

1. Introduction ▷ 2
2. Rational or Finite-State Transduction ▷ 6
3. Stochastic Finite-State Transducers ▷ 11
4. Error Correcting ▷ 20
5. Sequential Transduction ▷ 26
6. Subsequential Transduction: Introduction to the “OSTI” Algorithm ▷ 32
7. Bibliography ▷ 36
Subsequential Transduction

[Berstel,79]

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

- $\tau' = (Q, X, Y, q_0, E)$ is a Sequential Transducer
- $\sigma : Q \rightarrow Y^*$ is a state output (partial) function
- For each input string $x$, the output string $y$ is obtained by concatenating $\sigma(q)$ to $\tau'(x)$, where $q$ is the last state reached through the analysis of $x$ by $\tau'$; i.e.:
  \[ y = \tau(x) = \tau'(x)\sigma(q) \]

PROPERTIES:

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

Subsequential Transducers (intuitive concept)

- **Deterministic Finite State Networks** which accept sentences from an input language and produce sentences of an output language.
- In addition to input symbols, output strings are assigned to the edges.
- Output strings are also assigned to final states.
- **SST operation 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

y / triangle and

triangulo / λ

triangle
cuadrado / λ

cuadradó / λ

square

y / square and

grande / large triangle
grande / large square

λ
```
Learning SSTs: the OSTI Algorithm

[Oncina, 91-93]

SSTs can be learned from training examples using the Onward Subsequential Transducer Inference Algorithm (OSTIA).

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

Example:

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

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


Pattern Recognition approaches to Machine Translation
E. Vidal and F. Casacuberta
Pattern Recognition and Human Language Technology Group
Departament de Sistemes Informàtics i Computació
Institut Tecnològic d’Informàtica
Universitat Politècnica de València

5: Phrase-based Models and Alignment Templates

Francisco Casacuberta Nolla
fcn@iti.upv.es

24-28 January 2005

Index

1 Beyond word models ▷ 2
2 Phrase-based models ▷ 9
3 Alignment templates ▷ 47
4 Phrases and finite-state transducers ▷ 58
5 Using linguistic knowledge ▷ 66
6 Bibliography ▷ 72
Index

1 Beyond word models ➤ 2
2 Phrase-based models ➤ 9
3 Alignment templates ➤ 47
4 Phrases and finite-state transducers ➤ 58
5 Using linguistic knowledge ➤ 66
6 Bibliography ➤ 72

Exemple of word alignments

taxi . . . . . ■ . . . .
un . . . . . ■ . . . .
pídame ■ ■ ■ ■ . . . .
, . . . . . ■ . .
favor . . . . . . ■ .
por . . . . . . ■ .
could you ask for a taxi, please?
Segment alignment

SINGLE-WORD ALIGNMENTS: only model the correspondence between words.

Alternative:

SEGMENT ALIGNMENTS: modelling the correspondences between word segments.
Beyond word-based models

- The basic assumption in the current word-based models: Each source word is generated by only one target word.

- This assumption does not correspond to the nature of natural language. In some cases, it is necessary to know the context.

- Solutions:
  - Context-dependent dictionaries (previous talk). The basic unit is the word.
  - Word sequences:
    - Alignment templates: A sequence of source (classes of) words is aligned with a sequence of target (classes of) words. Inside the templates there are word-to-word correspondences. The basic unit is the word.
    - Phrase-based models: A sequence of source words is aligned with a sequence of target words. The basic unit is the phrase.

---

1By “phrase” we will mean a possible word sequence.

F. Casacuberta – DSIC-ITI-UPV 24-28 January 2005 5: 6

Word sequences

Alignment templates

Bilingual phrases
Phrase-based models

The statistical dictionaries of single word pairs are substituted by statistical dictionaries of *bilingual phrases*.

Bilingual phrases are related with a bilingual segmentation.

- Problem: The generalisation capability, since only sequences of segments that have been seeing in the training corpus are accepted.
- Problem: The selection of adequate bilingual phrases.

Index

1 Beyond word models ▷ 2
   ○ 2 Phrase-based models ▷ 9
3 Alignment templates ▷ 47
4 Phrases and finite-state transducers ▷ 58
5 Using linguistic knowledge ▷ 66
6 Bibliography ▷ 72
An example

y: could you ask for a taxi, please?

<table>
<thead>
<tr>
<th>y</th>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation</td>
<td>µ</td>
<td>µ₁</td>
<td>µ₂</td>
<td>µ₃</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permutation</td>
<td>α</td>
<td>α₁ = 2</td>
<td>α₂ = 3</td>
<td>α₃ = 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Translation</td>
<td>x</td>
<td>por favor,</td>
<td>pidame</td>
<td>un taxi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Segmentation</td>
<td>γ</td>
<td>γ₁</td>
<td>γ₂</td>
<td>γ₃</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

x: por favor, pidame un taxi.

General framework

- Let \( K \) be the number of segments in \( x \) and in \( y \),

- Segmentation of the target sentence
  
  \[ \mu : \{1, \ldots, K\} \rightarrow \{1, \ldots, I\} : \mu_k \geq \mu_{k-1} \quad 1 < k \leq K \quad \& \quad \mu_K = I \quad (\mu_0 = 0) \]

- Segmentation of the source sentence
  
  \[ \gamma : \{1, \ldots, K\} \rightarrow \{1, \ldots, J\} : \gamma_k \geq \gamma_{k-1} \quad 1 < k \leq K \quad \& \quad \gamma_K = J \quad (\gamma_0 = 0) \]

- Segment alignment (Permutation):
  
  \[ \alpha : \{1, \ldots, K\} \rightarrow \{1, \ldots, K\} : \alpha(k) = \alpha(k') \quad \text{iff} \quad k = k' \]
Monotone vs. no monotone alignments

**NO MONOTONE ALIGNMENT**

\[
\Pr(x|y) \approx P(x|y) = p(J|I) \cdot \sum_k \sum_{\alpha_k} \sum_{\gamma_k} \prod_{k=1}^K p(\alpha_k|\alpha_{k-1}) \cdot p(x_{\gamma_{\alpha_k-1}}|y_{\mu_{k-1}}) 
\]

**MONOTONE ALIGNMENT** ⇒ \( \alpha_k = k \)

\[
\Pr(x|y) \approx \hat{P}(x|y) = p(J|I) \cdot \max_k \max_{\alpha_k} \max_{\gamma_k} \prod_{k=1}^K p(\alpha_k|\alpha_{k-1}) \cdot p(x_{\gamma_{\alpha_k-1}}|y_{\mu_{k-1}}) 
\]

**Maximum approaches**

**NO MONOTONE ALIGNMENT**

\[
\Pr(x|y) \approx \hat{P}(x|y) = p(J|I) \cdot \max_k \max_{\alpha_k} \max_{\gamma_k} \prod_{k=1}^K p(\alpha_k|\alpha_{k-1}) \times p(x_{\gamma_{\alpha_k-1}}|y_{\mu_{k-1}}) 
\]

**MONOTONE ALIGNMENT** ⇒ \( \alpha_k = k \)

\[
\Pr(x|y) \approx \hat{P}(x|y) = p(J|I) \cdot \max_k \max_{\alpha_k} \max_{\gamma_k} \prod_{k=1}^K p(x_{\gamma_{k-1}}|y_{\mu_{k-1}}) 
\]
**Formal derivation (I)**

\[
\hat{y} = \arg\max_y \Pr(y|x) = \arg\max_y \Pr(y) \cdot \Pr(x|y)
\]

\[
\Pr(x | y_1^I) = \Pr(J | y_1^I) \cdot \Pr(x | y_1^I, J) = \Pr(J | y_1^I) \cdot \sum_K \Pr(K | y_1^I, J) \cdot \Pr(x | y_1^I, J, K)
\]

**Segmentation of target sentences**

\[
\mu : \{1, \ldots, K\} \rightarrow \{1, \ldots, I\} : \mu_k \geq \mu_{k-1} \quad 1 < k \leq K \quad \& \quad \mu_K = I \quad (\mu_0 = 0)
\]

\[
\text{Number of segments in } y: K = 3
\]

**An example (I)**

**x**: por favor, pídame un taxi

**y**: could you ask for a taxi, please?

\[
y \quad \text{could you ask for a taxi, please?} \\
1 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 = I
\]

**Segmentation of y**: \( \mu \)

\[
y \quad \text{could you ask for a taxi, please?} \\
1 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 = I
\]

\[
\mu \quad \mu_1 \quad \mu_2 \quad \mu_3
\]
Formal derivation (II)

\[ \Pr(x | y^I_1) = \Pr(J | y^I_1) \cdot \sum_K \sum_{\mu^K_1} \Pr(K | y^I_1, J) \cdot \Pr(\mu^K_1 | y^I_1, J, K) \cdot \Pr(x^I_1 | y^I_1, J, K, \mu^K_1) \]

Permutation of target segments:

\[ \alpha : \{1, \ldots, K\} \rightarrow \{1, \ldots, K\} : \alpha(k) = \alpha(k') \text{ iff } k = k' \]

\[ \Pr(x^I_1 | y^I_1, J, K, \mu^K_1) = \sum_{\alpha^K_1} \Pr(x^I_1, \alpha^K_1 | y^I_1, J, K, \mu^K_1) \]

\[ = \sum_{\alpha^K_1} \Pr(\alpha^K_1 | y^I_1, J, K, \mu^K_1) \cdot \Pr(x^I_1 | y^I_1, J, K, \mu^K_1, \alpha^K_1) \]

An example (II)

x: por favor, pidame un taxi

y: could you ask for a taxi, please?

Segmentation of y: \( \mu \)

<table>
<thead>
<tr>
<th>( y )</th>
<th>i</th>
<th>could</th>
<th>you</th>
<th>ask</th>
<th>for</th>
<th>a</th>
<th>taxi</th>
<th>,</th>
<th>please</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9=I</td>
<td></td>
</tr>
<tr>
<td>( \mu )</td>
<td>( \alpha_1 = 2 )</td>
<td>( \mu_1 )</td>
<td>( \alpha_2 = 3 )</td>
<td>( \mu_2 )</td>
<td>( \alpha_3 = 1 )</td>
<td>( \mu_3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Permutation of segments in y: \( \alpha \)

<table>
<thead>
<tr>
<th>( y )</th>
<th>i</th>
<th>please</th>
<th>?</th>
<th>could</th>
<th>you</th>
<th>ask</th>
<th>for</th>
<th>a</th>
<th>taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
An example (III)

x: por favor, pídame un taxi  
y: could you ask for a taxi, please?

Segmentation of x: $\gamma$

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Segments of x

x | por favor, | pídame | un taxi |

Segmentation of y: $\mu$

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_1$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>$\mu_3$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Permutation of segments in y: $\alpha$

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Pr($x_1^J | y_1^I, J, K, \mu_1^K, \alpha_1^K$) = $\sum_{\gamma_1^K} \Pr(x_1^J | y_1^I, J, K, \mu_1^K, \alpha_1^K)$

Pr($x_1^J | y_1^I, J, K, \mu_1^K, \alpha_1^K$) = $\sum_{\gamma_1^K} \Pr(\gamma_1^K | y_1^I, J, K, \mu_1^K, \alpha_1^K) \cdot \Pr(x_1^J | y_1^I, J, K, \mu_1^K, \alpha_1^K, \gamma_1^K)$
### Summary

\[
\Pr(x | y_1^I) = \Pr(J | y_1^I) \cdot \sum_K \frac{\sum_{\mu_1^K} \Pr(K | y_1^I, J) \cdot \Pr(\mu_1^K | y_1^I, J, K)}{\Pr(\mu_1^K | y_1^I, J, K)}.
\]

\[
\Pr(\alpha_1^K | y_1^I, J, \mu_1^K) \cdot \Pr(\gamma_1^K | y_1^I, J, K, \mu_1^K, \alpha_1^K) \cdot \Pr(x_1^J | y_1^I, J, K, \mu_1^K, \alpha_1^K, \gamma_1^K)
\]

\[
\Pr(K | y_1^I, J) \approx p(K | I, J)
\]

\[
\Pr(J | y_1^I) \approx p(J | I)
\]

\[
\Pr(\mu_k | y_1^I, J, \mu_1^{k-1}) \approx p(\mu_k | I)
\]

\[
\Pr(\alpha_k | y_1^I, J, \mu_1^K, \alpha_1^{k-1}) \approx p(\alpha_k | \alpha_{k-1})
\]

\[
\Pr(\gamma_{\alpha_k} | y_1^I, J, K, \mu_1^K, \alpha_1^K, \gamma_{\alpha_1}, \ldots, \gamma_{\alpha_{k-1}}) \approx p(\gamma_{\alpha_k} | I)
\]

\[
\Pr(x_{\gamma_{\alpha_k}^{k+1}}^J | y_1^I, J, K, \mu_1^K, \alpha_1^K, \gamma_1^K, x_{\gamma_{\alpha_1}^{k+1}}^J, \ldots, x_{\gamma_{\alpha_{k-1}}^{k+1}}^J) \approx p(x_{\gamma_{\alpha_k}^{k+1}}^{\mu_{k-1}^{k+1}} | y_{\mu_{k-1}^{k+1}}^{\mu_k})
\]
Another approach

\[ \Pr(J | y_1^I) \approx p(J|I) \]
\[ \Pr(K | y_1^I, J) \approx p(K|I, J) \]
\[ \Pr(\mu_k | y_1^I, J, K, \mu_1^{k-1}) \approx p(\mu_k|I, \mu_{k-1}) \]
\[ \Pr(\alpha_k | y_1^I, J, K, \mu_1^K, \alpha_1^{k-1}) \approx p(\alpha_k|\alpha_{k-1}) \]
\[ \Pr(\gamma_{\alpha_k} | y_1^I, J, K, \mu_1^K, \alpha_1^{K}, \gamma_{\alpha_1}, \ldots, \gamma_{\alpha_{k-1}}) \approx p(\gamma_{\alpha_k}|J, \mu_k, \gamma_{\alpha_{k-1}}) \]
\[ \Pr(x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}-1+1} | y_1^I, J, K, \mu_1^K, \alpha_1^{K}, \gamma_{\alpha_1}, \ldots, x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}-1+1}) \approx p(x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}-1+1} | y_{\mu_k}^{\mu_k}) \]

An extra approach

\[ \Pr(J | y_1^I) \approx p(J|I) \]
\[ \Pr(K | y_1^I, J) \approx p(K|I, J) \]
\[ \Pr(\mu_k | y_1^I, J, K, \mu_1^{k-1}) \approx p(\mu_k|I, y_{\mu_{k-1}}^{y_{\mu_k}}) \]
\[ \Pr(\alpha_k | y_1^I, J, K, \mu_1^K, \alpha_1^{k-1}) \approx p(\alpha_k|\alpha_{k-1}) \]
\[ \Pr(\gamma_{\alpha_k} | y_1^I, J, K, \mu_1^K, \alpha_1^{K}, \gamma_{\alpha_1}, \ldots, \gamma_{\alpha_{k-1}}) \approx \frac{p(\gamma_{\alpha_k}, x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}-1}, x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}})}{\sum_{s, s'} p(\gamma_{\alpha_k}, s, s')} \]
\[ \Pr(x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}-1+1} | y_1^I, J, K, \mu_1^K, \alpha_1^{K}, \gamma_{\alpha_1}, \ldots, x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}-1+1}) \approx p(x_{\gamma_{\alpha_k}}^{\gamma_{\alpha_k}-1+1} | y_{\mu_k}^{\mu_k}) \]
**Encore un plus**

\[
\Pr(J \mid y^I_1) \approx p(J \mid I)
\]

\[
\Pr(K \mid y^I_1, J) \approx p(K \mid I, J)
\]

\[
\Pr(\mu_k \mid y^I_1, J, K, \mu_k^{1-1}) \approx p(\mu_k \mid I, \mu_{k-1}, y^{\mu_k-2+1})
\]

\[
\Pr(\alpha_k \mid y^I_1, J, K, \mu_k^{1-1}) \approx p(\alpha_k \mid k, K) \cdot \prod_{k=1}^{K} (1 - \delta(\alpha_k, \alpha_i))
\]

\[
\Pr(\gamma_{\alpha_k} \mid y^I_1, J, K, \mu_k^{1-1}, \alpha_{1}, \cdots, \gamma_{\alpha_k-1}) \approx p(\gamma_{\alpha_k} \mid \gamma_{\alpha_k-1} \mid K, \mu_k - \mu_{k-1})
\]

\[
\Pr(x_{\gamma_{\alpha_k-1}+1} \mid y^I_1, J, K, \mu_k^{1-1}, \alpha_{1}, \cdots, x_{\gamma_{\alpha_k-1}+1}) \approx p(x_{\gamma_{\alpha_k-1}+1} \mid y^{\mu_k}_{\mu_k-1+1})
\]

**Monotone vs. no monotone alignments (I)**

**NO MONOTONE ALIGNMENT**

\[
\Pr(x \mid y^I_1) \approx P(x \mid y^I_1) = p(J \mid I) \cdot \sum_K \sum_{\mu_k^{1-1}} \sum_{\gamma_1^K} \sum_{\gamma_1^K} p(K \mid I, J) \cdot p(\mu_k \mid I) \cdot p(\alpha_k \mid \alpha_{k-1}) \cdot p(\gamma_{\alpha_k} \mid I) \cdot p(x_{\gamma_{\alpha_k-1}+1} \mid y^{\mu_k}_{\mu_k-1+1})
\]

**MONOTONE ALIGNMENT** \(\Rightarrow\) \(\alpha_k = k\)

\[
\Pr(x \mid y^I_1) \approx P(x \mid y^I_1) = p(J \mid I) \cdot \sum_K \sum_{\mu_k^{1-1}} \sum_{\gamma_1^K} p(K \mid I, J) \cdot \prod_{k=1}^{K} p(\mu_k \mid I) \cdot p(\gamma_k \mid J) \cdot p(x_{\gamma_{k-1}+1} \mid y^{\mu_k}_{\mu_k-1+1})
\]
Monotone vs. no monotone alignments (II)

**NO MONOTONE ALIGNMENT**

\[
Pr(x \mid y^1) \approx P(x \mid y^1) = P(J \mid I) \cdot \sum_{K} \sum_{\mu_k} \sum_{\gamma_k} p(K \mid I, J) \cdot \prod_{k=1}^{K} p(\mu_k \mid I, \mu_{k-1}, y_{\mu_{k-2}+1}).
\]

\[
p(\alpha_k \mid \alpha_{k-1} = k, K) \cdot p(x_{\gamma_{\alpha_{k-1}+1}} \mid y_{\mu_{k-1}+1}) \cdot \prod_{l=1}^{k} (1 - \delta(\alpha_k, \alpha_l)) \cdot p(\gamma_{\alpha_k} - \gamma_{\alpha_{k-1}} \mid K, \mu_k - \mu_{k-1}).
\]

**MONOTONE ALIGNMENT**

\[
Pr(x \mid y^1) \approx P(x \mid y^1) = P(J \mid I) \cdot \sum_{K} \sum_{\mu_k} \sum_{\gamma_k} p(K \mid I, J) \cdot \prod_{k=1}^{K} p(\mu_k \mid I, \mu_{k-1}, y_{\mu_{k-2}+1}) \cdot p(x_{\gamma_{k-1}+1} \mid y_{\mu_{k-1}+1}) \cdot \prod_{l=1}^{k} p(\gamma_k - \gamma_{k-1} \mid K, \mu_k - \mu_{k-1})
\]

Pattern Recognition approaches to Machine Translation

Statistical Alignment Models

Monotone phrase-based models

- Uniform distributions are assumed for \(p(J \mid I), p(K \mid I, J), p(\mu_k \mid I)\) and \(p(\gamma_k \mid J)\),

\[
P(x \mid y^1) \propto \sum_{K} \sum_{\mu_k} \sum_{\gamma_k} \prod_{k=1}^{K} p(x_{\gamma_{k-1}+1} \mid y_{\mu_{k-1}+1})
\]

- The sums are approximate by maximizations.

\[
P(x \mid y^1) \approx \max_{K} \max_{\mu_k} \max_{\gamma_k} \prod_{k=1}^{K} p(x_{\gamma_{k-1}+1} \mid y_{\mu_{k-1}+1})
\]
Learning phrase-based models

- Models
  - Learning monotone phrase-based models
  - Learning nonmonotone phrase-based models

- Phrase-based units
  - Training with a sentence-aligned corpus.
  - Training with a word-aligned corpus.

Learning monotone phrase-based models*

Training with a sentence-aligned corpus.

Given a sentence-aligned corpus \( \mathcal{T} \) of pairs of sentences \((x, y)\), the maximum likelihood criterium tries to estimate the parameters \( p(\tilde{x} \mid \tilde{y}) \) that maximize:

\[
\prod_{(x, y) \in \mathcal{T}} P(x \mid y)
\]

subject to: \( \sum_{\tilde{x}} p(\tilde{x} \mid \tilde{y}) = 1 \) for each target segment \( \tilde{y} \)

By applying an EM procedure:

\[
p(\tilde{x} \mid \tilde{y}) = \lambda_{\tilde{y}} \cdot \sum_{(x, y) \in \mathcal{T}} \sum_{K, \mu_1^K, \gamma_1^K} \left( \prod_{k=1}^{K} p(x_{\gamma_k}^{\gamma_{k-1}+1} \mid y_{\mu_{k-1}+1}^{\mu_k}) \cdot \sum_{l=1}^{K} \delta(\tilde{x} = x_{\gamma_l}^{\gamma_{l-1}+1}) \cdot \delta(\tilde{y} = y_{\mu_{l-1}+1}^{\mu_l}) \right)
\]

where \( \lambda_{\tilde{y}} \) is a normalization factor and \( \delta \): \( \delta(true) = 1 \) \( \delta(false) = 0 \).

*The slides on phrase-based models are modified versions of some material supplied by Jesús Tomás.
Learning monotone phrase-based models

Statistical Alignment Models

Training with a word-aligned corpus.

Given a sentence-aligned corpus $T$, 

- A word-aligned corpus is generated using the GIZA++ toolkit with $T$
  http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html
- A set of bilingual word sequences from the word aligned corpus is extracted.
- The parameters of the phrase-model are estimated.

Extracting bilingual word sequences.

For each $x, y \in T$, aligned by $a$,

$$BP_1(x, y, a) = \left\{ (x_{j_1}^{j_2}, y_{i_1}^{i_2}) : \forall j : j_1 \leq j \leq j_2; \exists i : i_1 \leq i \leq i_2 : a(j) = i \right\}$$

$$BP_2(x, y, a) = \left\{ (x_{j_1}^{j_2}, y_{i_1}^{i_2}) : \forall j : j_1 \leq j \leq j_2; (i_1 \leq a(j) \leq i_2) \lor (a(j) = 0) \right\}$$

$$BP_3(x, y, a) = \left\{ (x_{j_1}^{j_2}, y_{i_1}^{i_2}) : \forall j : j_1 \leq j \leq j_2; (i_1 \leq a_x y(j) \leq i_2) \lor (a(j) = 0) \right\}$$
Learning monotone phrase-based models

Extracting bilingual multiword sequences: an example

\[ x: \text{configuration program} \]
\[ y: \text{programa de configuración} \]
\[ a: \begin{array}{ccc} 2 & 0 & 1 \end{array} \]

- \( BP_1 = \{ \text{configuration-configuración}, \text{program-programa} \} \)
- \( BP_2 = \{ \text{configuration-configuración}, \text{program-programa}, \text{configuration-de configuración}, \text{program-programa de, configuration-programa de configuración} \} \)
- \( BP_3 = \{ \text{configuration program-programa de configuración} \} \)

Estimating the parameters.

By relative frequencies, for each pair of segments \((x, y)\):

\[
p(x \mid y) = \frac{N(x, y)}{N(y)}
\]

where \( N(y) \) denotes the number of times that phrase \( \tilde{y} \) has appeared, and \( N(x, y) \) is the number of times that the bilingual phrase \( (\tilde{x}, \tilde{y}) \) has appeared.

A refinement: the combination of the method based on a sentence-aligned corpus and one of the techniques for the bilingual multiword sequences:

\[
p(\tilde{x} \mid \tilde{y}) = \lambda_{\tilde{y}} \cdot \sum_{(x, y) \in T} p_T \cdot \sum_{K, \mu_1, \gamma_1} \left( \prod_{k=1}^{K} p(x_{\gamma_k} \mid y_{\mu_{k-1}+1}) \cdot \delta((\tilde{x}, \tilde{y}) \in BP(T)) \right)
\]
Learning no-monotone phrase-based models

- The procedures for estimating the models parameters are similar to the ones for monotone models.

- For the distortion model, $p(\alpha_k | \alpha_{k-1})^\gamma$:

\[
p(\alpha_k | \alpha_{k-1}) = p_0^{\gamma_k - \gamma_{k-1}},
\]

where $p_0$ is a parameter to be adjusted using a validation set.


Search algorithms for monotone phrase-based models

- Basic idea is to generate partial hypothesis about the target sentence in an incremental way.

- Each of these hypothesis is composed by a prefix of the target sentence, a subset of source positions that have been aligned with the positions of the prefix of the target sentence and a score.

- New hypothesis can be generated for a previous hypothesis by adding a target word to the prefix of the target sentence that is the translation of a source(s) word(s) that is (are) not translated yet.

The adopted search algorithm for phrase-based models is the \textit{multi-stack-decoding algorithm}.
Search algorithms for monotone phrase-based models

Given a source sentence $x$, a hypothesis is a tuple

$$\left( x^j_1, y^i_1, S(x^j_1, y^i_1) = P(y^i_1) \cdot P(x^j_1 | y^i_1) \right)$$

where

$$P(y^i_1) = \prod_{i' = 1}^{i} p(y_{i'} | y_{i'-n+1})$$

and

$$P(x^j_1 | y^i_1) = \max_{K} \max_{\mu_k} \prod_{k=1}^{K} p(x_{\gamma_k+1}^j | y_{\mu_k+1}^i)$$

with $\gamma_k = j$ and $\mu_k = i$

- The initialization consists on building a hypothesis for the empty target and empty source prefixes and a score of 1.0.

- The algorithm selects a hypothesis $(x^j_1, y^i_1, S(x^j_1, y^i_1))$ of each stack and for each bilingual segment $(\bar{x}, \bar{y})$ with $x^j_{j+1} \equiv \bar{x}$, a new hypothesis is created $(x^j_1 \bar{x}, y^i_1 \bar{y}, S(x^j_1 \bar{x}, y^i_1 \bar{y}))$

$$S(x^j_1 \bar{x}, y^i_1 \bar{y}) = S(x^j_1, y^i_1) \cdot \prod_{l=i+1}^{i+|\bar{y}|} p(y_l | y_{l-n+1}^i) \cdot p(\bar{x} | \bar{y}) \quad (1)$$

Each new hypothesis, $(x^j_1 \bar{x}, y^i_1 \bar{y}, S(x^j_1 \bar{x}, y^i_1 \bar{y}))$ will be stored in the stack associated to the source prefixes of length $j + |\bar{x}|$.
Search algorithms for no-monotone search algorithm.

- The procedure is quite similar to the monotone search algorithm,

- A hypothesis consists on a prefix of the target sentence, a subset of source positions and a score with the partial contributions of the target language model and translation model.

- The implementation requires a stack for each possible subset of source positions and consequently, the computational cost can be very high.

Experimental results

Corpora

“El Periódico”: From a bilingual newspaper (Spanish to Catalan)

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>Catalan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train:</td>
<td>Sentence pairs</td>
<td>643,961</td>
</tr>
<tr>
<td></td>
<td>Running words (Kwords)</td>
<td>7,180</td>
</tr>
<tr>
<td></td>
<td>Vocabulary (Kwords)</td>
<td>129</td>
</tr>
<tr>
<td>Test:</td>
<td>Sentences</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td>Running words</td>
<td>4,316</td>
</tr>
</tbody>
</table>
Experimental results

Corpora

**XRCE**: From Xerox printer manuals (English to and from Spanish, French and German)

<table>
<thead>
<tr>
<th></th>
<th>En</th>
<th>Sp</th>
<th>En</th>
<th>Ge</th>
<th>En</th>
<th>Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train: Sentence pairs</td>
<td>56K</td>
<td>49K</td>
<td></td>
<td></td>
<td></td>
<td>53K</td>
</tr>
<tr>
<td>Running words</td>
<td>665K</td>
<td>753K</td>
<td>633K</td>
<td>696K</td>
<td>587K</td>
<td>534K</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>8K</td>
<td>11K</td>
<td>8K</td>
<td>10K</td>
<td>8K</td>
<td>19K</td>
</tr>
<tr>
<td>Test: Sentence pairs</td>
<td>1,125</td>
<td>984</td>
<td>996</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running words</td>
<td>8K</td>
<td>10K</td>
<td>11K</td>
<td>12K</td>
<td>12K</td>
<td>12K</td>
</tr>
<tr>
<td>Test perplexity</td>
<td>48</td>
<td>33</td>
<td>51</td>
<td>87</td>
<td>73</td>
<td>52</td>
</tr>
</tbody>
</table>

**EU**: Bulletin of the European Union (English to and from Spanish, French and German)

<table>
<thead>
<tr>
<th></th>
<th>En</th>
<th>Sp</th>
<th>En</th>
<th>Ge</th>
<th>En</th>
<th>Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train: Sentence pairs</td>
<td>214K</td>
<td>223K</td>
<td>215K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running words</td>
<td>5.9M</td>
<td>6.6M</td>
<td>6.5M</td>
<td>6.1M</td>
<td>6.0M</td>
<td>6.6M</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>84K</td>
<td>97K</td>
<td>87K</td>
<td>152K</td>
<td>85K</td>
<td>91K</td>
</tr>
<tr>
<td>Test: Sentence pairs</td>
<td>800</td>
<td>800</td>
<td>800</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running words</td>
<td>2K</td>
<td>25K</td>
<td>22K</td>
<td>21K</td>
<td>22K</td>
<td>24K</td>
</tr>
<tr>
<td>Test perplexity</td>
<td>47</td>
<td>39</td>
<td>47</td>
<td>71</td>
<td>48</td>
<td>38</td>
</tr>
</tbody>
</table>
Experimental results

Corpora

Hansards: Proceedings of Canadian Parliament (French to English)

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train: Sentence pairs</td>
<td>137,381</td>
<td></td>
</tr>
<tr>
<td>Running words (Kwords)</td>
<td>1,941</td>
<td>2,130</td>
</tr>
<tr>
<td>Vocabulary (Kwords)</td>
<td>29.5</td>
<td>37.5</td>
</tr>
<tr>
<td>Test: Sentences</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>Running words</td>
<td>2,633</td>
<td>2,805</td>
</tr>
</tbody>
</table>

Assessment

- **Word error rate (WER)**: The minimum number of substitution, insertion and deletion operations needed to convert the word string hypothesized by the translation system into a given single reference word string.

- **Multi reference WER (mWER)**: Similar to WER, but for each source test sentence there are more than one target sentences as references.

- **BiLingual Evaluation Understudy (BLEU)**: it is based on the $n$-grams of the hypothesized translation that occur in the reference translations. The BLEU metric ranges from 0.0 (worst score) to 1.0 (best score).
### Effect of the size of the segment length (MSL)

#### “El Periódico” task.

<table>
<thead>
<tr>
<th>MSL</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER</td>
<td>12.1</td>
<td>10.6</td>
<td>10.5</td>
</tr>
<tr>
<td>Parameters</td>
<td>2.0M</td>
<td>7.0M</td>
<td>14.5M</td>
</tr>
</tbody>
</table>

#### XRCE task.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>En-Es</th>
<th>Es-En</th>
<th>En-Fr</th>
<th>Fr-En</th>
<th>En-De</th>
<th>De-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP1</td>
<td>45.7</td>
<td>28.6</td>
<td>54.2</td>
<td>52.4</td>
<td>65.1</td>
<td>55.6</td>
</tr>
<tr>
<td>BP2</td>
<td>26.4</td>
<td>27.4</td>
<td>53.6</td>
<td>52.2</td>
<td>64.1</td>
<td>54.3</td>
</tr>
<tr>
<td>BP3</td>
<td>25.4</td>
<td>27.3</td>
<td>54.0</td>
<td>52.5</td>
<td>64.9</td>
<td>53.9</td>
</tr>
</tbody>
</table>

#### Monotone vs. non monotone search. XRCE task.

<table>
<thead>
<tr>
<th>Search</th>
<th>En-Es</th>
<th>Es-En</th>
<th>En-Fr</th>
<th>Fr-En</th>
<th>En-De</th>
<th>De-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotone</td>
<td>28.5</td>
<td>30.9</td>
<td>51.4</td>
<td>51.6</td>
<td>66.4</td>
<td>54.1</td>
</tr>
<tr>
<td>No monotone</td>
<td>28.0</td>
<td>31.6</td>
<td>52.0</td>
<td>51.3</td>
<td>66.4</td>
<td>54.0</td>
</tr>
</tbody>
</table>

### Some results

#### Different methods for building segments. XRCE task.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>En-Es</th>
<th>Es-En</th>
<th>En-Fr</th>
<th>Fr-En</th>
<th>En-De</th>
<th>De-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP1</td>
<td>45.7</td>
<td>28.6</td>
<td>54.2</td>
<td>52.4</td>
<td>65.1</td>
<td>55.6</td>
</tr>
<tr>
<td>BP2</td>
<td>26.4</td>
<td>27.4</td>
<td>53.6</td>
<td>52.2</td>
<td>64.1</td>
<td>54.3</td>
</tr>
<tr>
<td>BP3</td>
<td>25.4</td>
<td>27.3</td>
<td>54.0</td>
<td>52.5</td>
<td>64.9</td>
<td>53.9</td>
</tr>
</tbody>
</table>

#### Effect of the training set on the system performance. “El Periódico” task.

<table>
<thead>
<tr>
<th>Corpus size</th>
<th>5K</th>
<th>10K</th>
<th>20K</th>
<th>40K</th>
<th>80K</th>
<th>160K</th>
<th>320K</th>
<th>640K</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER</td>
<td>20.3</td>
<td>17.3</td>
<td>15.2</td>
<td>13.4</td>
<td>12.4</td>
<td>11.7</td>
<td>11.1</td>
<td>10.7</td>
</tr>
<tr>
<td>Parameters</td>
<td>0.1</td>
<td>0.2M</td>
<td>0.4M</td>
<td>0.7M</td>
<td>1.2M</td>
<td>2.1M</td>
<td>3.6M</td>
<td>7.0M</td>
</tr>
</tbody>
</table>
Comparison with other machine translation systems.

“El Periódico” task.

- **Salt** a knowledge-based machine translation systems supported by the Government of the Generalitat Valenciana (http://www.cultgva.es).

- **Incyta**, a knowledge-based commercial systems (http://www.incyta.com).

- **InterNOSTRUM**, a hybrid knowledge-based and finite-state translation system (http://www.internostrum.com).

<table>
<thead>
<tr>
<th>MT system</th>
<th>WER (%)</th>
<th>mWER (%)</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salt</td>
<td>9.9</td>
<td>6.6</td>
<td>0.866</td>
</tr>
<tr>
<td>Incyta</td>
<td>10.0</td>
<td>7.6</td>
<td>0.855</td>
</tr>
<tr>
<td>Phrase-based</td>
<td>10.7</td>
<td>7.8</td>
<td>0.857</td>
</tr>
<tr>
<td>InterNOSTRUM</td>
<td>11.9</td>
<td>8.5</td>
<td>0.837</td>
</tr>
</tbody>
</table>

Finally, a simple experiment was carried out with the HANSARD task. The result obtained was 64.9% of WER.

Index

1. Beyond word models ▶ 2
2. Phrase-based models ▶ 9
3. Alignment templates ▶ 47
4. Phrases and finite-state transducers ▶ 58
5. Using linguistic knowledge ▶ 66
6. Bibliography ▶ 72
Alignment templates


| ? | . | . | . | . | . | . | . | . |
| semana | . | . | . | . | . | . | . | . |
| por | . | . | . | . | . | . | . | . |
| individual | . | . | . | . | . | . | . | . |
| habitación | . | . | . | . | . | . | . | . |
| una | . | . | . | . | . | . | . | . |
| cuesta | . | . | . | . | . | . | . | . |
| Cuánto | . | . | . | . | . | . | . | . |

how | much | does | a | single | room | cost | Per | week | ? |

F. Casacuberta – DSIC-ITI-UPV 24-28 January 2005 5:48
Alignment templates


- Let $K$ be the number of segments in $x$ and in $y$,
- Segmentation of the target sentence
  \[
  \mu : \{1, \ldots, K\} \rightarrow \{1, \ldots, I\} : \mu_k \geq \mu_{k-1} \quad 1 < k \leq K \quad \& \quad \mu_K = I \quad (\mu_0 = 0)
  \]
  \[
  y^\mu_1 \Rightarrow \tilde{y}^\mu_1 ; \tilde{y}_k \equiv y_{\mu_k}^{\mu_k+1} = y_{\mu_k}^{\mu_k+1}, \ldots, y_{\mu_k}^I ; 1 \leq k \leq K
  \]
- Segmentation of the source sentence
  \[
  \gamma : \{1, \ldots, K\} \rightarrow \{1, \ldots, J\} : \gamma_k \geq \gamma_{k-1} \quad 1 < k \leq K \quad \& \quad \gamma_K = J \quad (\gamma_0 = 0)
  \]
  \[
  x^\gamma_1 \Rightarrow \tilde{x}^\gamma_1 ; \tilde{x}_k \equiv x_{\gamma_k}^{\gamma_k+1} = x_{\gamma_k}^{\gamma_k+1}, \ldots, x_{\gamma_k}^J ; 1 \leq k \leq K
  \]
- Segment alignment (Permutation):
  \[
  \alpha : \{1, \ldots, K\} \rightarrow \{1, \ldots, K\} : \alpha(k) = \alpha(k') \iff \ k = k'
  \]

Alignment templates

ALIGNMENT BETWEEN WORD GROUPS

\[
Pr(\tilde{x} \mid \tilde{y}) = \sum_\alpha Pr(\alpha, \tilde{x} \mid \tilde{y}) \approx \sum_\alpha \prod_{k=1}^{K} p(\alpha_k \mid \alpha_{k-1}) \cdot P(\tilde{x}_k \mid \tilde{y}_{\alpha_k})
\]

ALIGNMENT WITHIN WORD GROUPS

\[
P(\tilde{x}_k \mid \tilde{y}_l) = \sum_Z p(Z \mid \tilde{y}_l) \cdot p(\tilde{x}_k \mid Z, \tilde{y}_l)
\]

An ALIGNMENT TEMPLATE $z$ is a binary matrix with $I'$ rows and $J'$ columns: $z_{i,j} = 1$ if the pair $y_i$ and $x_j$ are aligned.

[ despertar ]
[ podrían ]
[ nos ]
[ ¿ ]

[could you wake us up]

F. Casacuberta – DSIC-ITI-UPV 24-28 January 2005 5: 50
Alignment templates


\[
P(\tilde{x}_k \mid \tilde{y}_l) \approx \sum_Z p(Z \mid \tilde{y}_l) \cdot \prod_{j=1}^{J'} \sum_{i=1}^{I'} a(i \mid j, z) \cdot l(\tilde{x}_{kj} \mid \tilde{y}_{li})
\]

\[
= \sum_Z p(Z \mid \tilde{y}_l) \cdot \prod_{j=1}^{J'} \sum_{i=1}^{I'} \sum_{i'} z_{ij} z_{i'j} \cdot l(\tilde{x}_{kj} \mid \tilde{y}_{li})
\]

Alignment templates: training and search


- **Training:**
  - Viterbi alignment \( x \rightarrow y \) and \( y \rightarrow x \).
  - Obtaining all template templates by considering all possible source-target word groups under the constraint that the words within the source/target word group are only aligned to words within the target/source word group.

- **Translation:**
  - Computing all possible segmentations of the source sentence into word groups.
  - Computing all possible alignments between word groups.
  - Computing all possible word alignments within the word group.
Results

(EUTRANS consortium Example-Based Language Translation Systems. Final Report. Deliverable D0.1c. 2000.)

EuTrans-I corpus (Spanish-English)

- Vocabulary: 680 Spanish words, and 513 English words.
- Training: 10,000 pairs (97,000/99,000 words).
- Test: 2,996 pairs (PP=3.3) (35,000/35,590 words).

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment templates (with manual categories)</td>
<td>2.5</td>
</tr>
<tr>
<td>Quasi-Monotone search</td>
<td>10.8</td>
</tr>
<tr>
<td>DP-search M2</td>
<td>13.9</td>
</tr>
</tbody>
</table>

FUB corpus (Italian-English)

- Vocabulary: 2,458 Italian words, and 1,701 English words.
- Training: 3,338 pairs (61,423/72,689 words).
- Test: 278 pairs (PP=31).

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment templates</td>
<td>23.8</td>
</tr>
<tr>
<td>Monotone search</td>
<td>29.3</td>
</tr>
</tbody>
</table>
Pattern Recognition approaches to Machine Translation

Statistical Alignment Models

Results

(H. Ney et al., Algorithms for statistical translation of spoken language. IEEE TSAP. 2000.)

Vermobil corpus (German-English)

- **Vocabulary**: 5,936 German words, and 3,505 English words.
- **Training**: 30,556 pairs (329,000/343,000 words).
- **Test**: 47 pairs (PP=43.7) (701/792 words).

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment templates</td>
<td>28.8</td>
</tr>
<tr>
<td>Inverted search</td>
<td>41.0</td>
</tr>
<tr>
<td>Monotone search</td>
<td>36.5</td>
</tr>
</tbody>
</table>

----

Vermobil corpus (German-English)

- **Vocabulary**: 7,940 German words, and 4,673 English words.
- **Training**: 58,332 pairs (519,523/549,921 words).
- **Test**: 5,069 (German → English) and 4,136 (English → German) sentences.

<table>
<thead>
<tr>
<th>Model</th>
<th>SER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic Transfer</td>
<td>62</td>
</tr>
<tr>
<td>Dialog Act Based</td>
<td>60</td>
</tr>
<tr>
<td>Example Based</td>
<td>51</td>
</tr>
<tr>
<td>Statistical</td>
<td>29</td>
</tr>
</tbody>
</table>
Index

1 Beyond word models ⊿ 2
2 Phrase-based models ⊿ 9
3 Alignment templates ⊿ 47
4 Phrases and finite-state transducers ⊿ 58
5 Using linguistic knowledge ⊿ 66
6 Bibliography ⊿ 72

Joint distributions (I)

\[ \hat{y} = \arg\max_y \Pr(y \mid x) = \arg\max_y \Pr(x, y) \]

Assuming monotone constraints:

\[
\Pr(x, y) = \Pr(J, I) \cdot \Pr(x^I_1, y^I_1 \mid J, I)
\]

\[
= \Pr(J, I) \cdot \sum_K \Pr(K \mid J, I) \cdot \Pr(x^I_1, y^I_1 \mid J, I, K)
\]

\[
= \Pr(J, I) \cdot \sum_K \Pr(K \mid J, I) \cdot \sum_{\gamma^K_1, \mu^K_1} \Pr(x^I_1, y^I_1, \gamma^K_1, \mu^K_1 \mid J, I, K)
\]


Joint distributions (II)

\[ \Pr(x, y) = \Pr(J, I) \cdot \sum_K \Pr(K \mid J, I) \cdot \sum_{\gamma^K, \mu^K} \Pr(x^I, y^I, \gamma^K, \mu^K \mid J, I, K) \]

\[ = \Pr(J, I) \cdot \sum_K \Pr(K \mid J, I) \cdot \sum_{\gamma^K, \mu^K} \Pr(\gamma^K, \mu^K \mid J, I, K) \cdot \Pr(x^I, y^I \mid J, I, K, \gamma^K, \mu^K) \]

\[ = \Pr(J, I) \cdot \sum_K \Pr(K \mid J, I) \cdot \prod_{k=1}^K \Pr(\gamma_k, \mu_k \mid J, I, K, \gamma_1^{k-1}, \mu_1^{k-1}) \cdot \Pr(x_1^{\gamma_k+1}, y_1^{\mu_k+1} \mid J, I, K, x_1^{\gamma_k-1}, y_1^{\mu_k-1}, \gamma^K, \mu^K) \]

Joint distributions (III)

\[ \Pr(x^I, y^I, \gamma^K, \mu^K \mid J, I, K) = \prod_{k=1}^K \Pr(\gamma_k, \mu_k \mid J, I, K, \gamma_1^{k-1}, \mu_1^{k-1}) \cdot \Pr(x_1^{\gamma_k+1}, y_1^{\mu_k+1} \mid J, I, K, x_1^{\gamma_k-1}, y_1^{\mu_k-1}, \gamma^K, \mu^K) \]

Assuming,

- \( \Pr(\gamma_k, \mu_k \mid J, I, K, \gamma_1^{k-1}, \mu_1^{k-1}) \approx \rho \)

- \( \Pr(x_1^{\gamma_k+1}, y_1^{\mu_k+1} \mid J, I, K, x_1^{\gamma_k-1}, y_1^{\mu_k-1}, \gamma^K, \mu^K) \approx \Pr(x_1^{\gamma_k+1}, y_1^{\mu_k+1} \mid x_1^{\gamma_k-1}, y_1^{\mu_k-1}) \)

\[ \Pr(x^I, y^I, \gamma^K, \mu^K \mid J, I, K) \approx \rho \cdot \prod_{k=1}^K \Pr(x_1^{\gamma_k+1}, y_1^{\mu_k+1} \mid x_1^{\gamma_k-1}, y_1^{\mu_k-1}) \]

Assuming n-grams,

\[ \Pr(x^I, y^I, \gamma^K, \mu^K \mid J, I, K) \approx \rho \cdot \prod_{k=1}^K \Pr(x_1^{\gamma_k+1}, y_1^{\mu_k+1} \mid x_1^{\gamma_k-1}, y_1^{\mu_k-1}) \]
An example (IV)

x: he hecho una reserva de una habitación doble .

y: I have made a reservation of a double room .

The lengths of x and y

<table>
<thead>
<tr>
<th>j</th>
<th>1 2 3 4 5 6 7 8 9=J</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>hecho una reserva de una habitación doble .</td>
</tr>
<tr>
<td>y</td>
<td>I have made a reservation of a double room</td>
</tr>
</tbody>
</table>

Number of segments: $K = 3$

An example (V)

x: he hecho una reserva de una habitación doble .

y: I have made a reservation of a double room .

Segmentation of x and y

<table>
<thead>
<tr>
<th>j</th>
<th>1 2 3 4 5 6 7 8 9=J</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>hecho una reserva de una habitación doble .</td>
</tr>
<tr>
<td>y</td>
<td>I have made a reservation of a double room</td>
</tr>
</tbody>
</table>

Phrases of x and y

<table>
<thead>
<tr>
<th>j</th>
<th>1 2</th>
<th>3 4 5 6 7 8 9=J</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>hecho</td>
<td>una reserva de una habitación doble .</td>
</tr>
<tr>
<td>y</td>
<td>I have made</td>
<td>a reservation of a double room</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>1 2 3 4 5 6 7 8 9 10=I</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>I have made a reservation of a double room</td>
</tr>
<tr>
<td>x</td>
<td>he hecho una reserva de una habitación doble .</td>
</tr>
</tbody>
</table>
Joint distributions (IV)

\[
\Pr(x^J_1, y^I_1 \mid J, I) \approx \rho \cdot \sum_{K} \Pr(K \mid J, I) \cdot \sum_{\gamma_{k}^{\mu_{k-1}}} \prod_{k=1}^{K} \Pr(x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k})
\]

\[
\approx \rho \cdot \max_{K} \Pr(K \mid J, I) \cdot \max_{\gamma_{k}^{\mu_{k-1}}} \prod_{k=1}^{K} \Pr(x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k})
\]

A simple case: GIATI (L_I): \( K = J, \gamma_j = j \) and the target segments can be empty

Remark: There can be test segmentations that are not in the training corpus!

Possible smoothings:

\[
\Pr(x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k}) = \begin{cases} 
\Pr(x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k}) & \gamma_j \neq j \\
\Pr(x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k}) & \gamma_j = j \\
\Pr(x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k} \mid -,-) 
\end{cases}
\]

Joint distributions and HMMs

\[
\Pr(x^J_1, y^I_1 \mid J, I) \approx \rho' \cdot \sum_{K} \prod_{k=1}^{K} \Pr(y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k}) \cdot \Pr(x_{\gamma_{k-1}+1}^{\gamma_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k})
\]

\[
\Pr(y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k}) \approx \Pr(y_{\mu_{k-1}+1}^{\mu_k} \mid y_{1}^{\mu_{k-1}}) \approx \prod_{l=\mu_{k-1}+1}^{\mu_k} \Pr(y_{l} \mid y_{1}^{\mu_{k-1}})
\]

\[
\Pr(x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k} \mid x_{\gamma_{k-1}+1}^{\gamma_k}, y_{\mu_{k-1}+1}^{\mu_k}) \approx \Pr(x_{\gamma_{k-1}+1}^{\gamma_k} \mid y_{\mu_{k-1}+1}^{\mu_k})
\]

\[
\Pr(x^J_1, y^I_1 \mid J, I) \approx \rho' \cdot \sum_{K} \prod_{k=1}^{K} \left( \Pr(x_{\gamma_{k-1}+1}^{\gamma_k} \mid y_{\mu_{k-1}+1}^{\mu_k}) \cdot \prod_{l=\mu_{k-1}+1}^{\mu_k} \Pr(y_{l} \mid y_{1}^{\mu_{k-1}}) \right)
\]
Chunking in statistical machine translation


1. Chunking the source sentence: generating sequence of chunks (with the corresponding source POS)

2. Reordering the source chunks.

3. Chunk mapping: generating the target POS of each chunk

4. Word translations: generating the target words.

Marginal improvements on a corpus of the European Parliament proceedings (German to English)
Bilingual chunking in statistical machine translation


\[
\text{argmax}_{\sigma, \tau, \alpha} \Pr(y, \sigma, \tau, \alpha \mid x)
\]

- \(\sigma\) = sequence of source chunks;
- \(\tau\) = sequence of target chunks;
- \(\alpha\) = alignment between chunks.

By introducing POS tagging of the source sentence \(\pi_x\) and POS tagging of the target sentence \(\pi_y\),

\[
\text{argmax}_{\sigma, \tau, \alpha, \pi_y, \pi_x} \Pr(\pi_x \mid x) \cdot \Pr(\sigma \mid \pi_x, x) \cdot \Pr(\pi_y \mid \sigma, \pi_x, x) \\
\cdot \Pr(y \mid \pi_y, \sigma, \pi_x, x) \cdot \Pr(\tau \mid y, \pi_y, \sigma, \pi_x, x) \cdot \Pr(\alpha \mid \tau, y, \pi_y, \sigma, \pi_x, x)
\]

Only results on chunk alignments.
Parsing in statistical machine translation


\[
Pr(y | x) = \sum_{\pi y} Pr(y, \pi y) \cdot Pr(x | y, \pi y) = \sum_{\pi y} Pr(\pi y) \cdot Pr(x | \pi y)
\]

\(\pi y\) is a parse tree of the target sentence \(y\)

Three operations for \(Pr(x | \pi y)\):

- Reordering some of the child nodes
- Inserting optional words
- Translating each target word by the corresponding source word

Some improvements on a Chinese to English newspaper task.

Other approaches


Index

1. Beyond word models ▷ 2
2. Phrase-based models ▷ 9
3. Alignment templates ▷ 47
4. Phrases and finite-state transducers ▷ 58
5. Using linguistic knowledge ▷ 66
6. Bibliography ▷ 72

Bibliography

Bibliography


Bibliografia


Pattern Recognition approaches to Machine Translation

F. Casacuberta and E. Vidal

Pattern Recognition and Human Language Technology Group
Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain

State-Merging Approaches

Enrique Vidal
evidal@iti.upv.es

January 2005

E. Vidal – ITI-UPV-DSIC

Index

1 Subsequential Transduction: “OSTI” Algorithm \(\rceil\) 2
2 Using input/output syntactic constraints: OSTIA-DR \(\rceil\) 29
3 OSTIA-DR: improving scalability \(\rceil\) 45
4 Bibliography \(\rceil\) 70
Index

1 Subsequential Transduction: “OSTI” Algorithm > 2
2 Using input/output syntactic constraints: OSTIA-DR > 29
3 OSTIA-DR: improving scalability > 45
4 Bibliography > 70

Sequential Transducers

A Sequential Transducer (ST) \( \tau \) is a 5-tuple \( \tau = (Q, X, Y, q_0, E) \):

- \( Q \): Finite set of States
- \( X, Y \): Input and output Alphabets
- \( q_0 \in Q \): Initial State
- \( E \subseteq Q \times X \times Y^* \times Q \): “Edges” or Transitions

- All the states are accepting
- Edges are deterministic:
  \( (q, a, u, r), (q, a, v, s) \in E \Rightarrow (u = v \land r = s) \)

PROPERTIES:

1. \( T_\tau \) is a function: \( X^* \to Y^* \)
2. STs \( \equiv \) Generalized Sequential Machines \( \supseteq \) (Mealy and Moore machines)
3. STs preserve prefixes: \( T_\tau(\lambda) = \lambda; \ T_\tau(uv) \in T_\tau(u)Y^* \)

“Property” 2 entails strict sequentiality, which can hardly be adequate in many cases of interest
Subsequential Transduction

[Berstel, 79]

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

- $\tau' = (Q, X, Y, q_0, E)$ is a Sequential Transducer
- $\sigma : Q \rightarrow Y^*$ is a state output (partial) function
- For each input string $x$, the output string $y$ is obtained by concatenating $\sigma(q)$ to $\tau'(x)$, where $q$ is the last state reached through the analysis of $x$ by $\tau'$; i.e.:

$$y = \tau(x) = \tau'(x)\sigma(q)$$

PROPERTIES:

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

Subsequential Transducers (intuitive concept)

- Deterministic Finite State Networks which accept sentences from an input language and produce sentences of an output language.

- In addition to input symbols, output strings are assigned to the edges.

- Output strings are also assigned to final states.

- SST operation 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:
Learning SSTs: the OSTI Algorithm

[Oncina, 91-93]

SSTs can be learned from training examples using the Onward Subsequential Transducer Inference Algorithm (OSTIA).

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

   Example:
   
   \[(un \text{ triángulo y un cuadrado}, \text{ a triangle and a square}),\]
   
   \[(un \text{ triángulo grande}, \text{ a large triangle}),\]
   
   \[(un \text{ cuadrado}, \text{ a square})\]

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

OSTIA State-Merging learning procedure

- The traversal of the tree follows a level by level order, typically using the lexicographic order of state names.

- Two kinds of State Merging:
  
  - Merging based on local conditions: involve only the two states under consideration. The most basic idea [Oncina, 91-93]:
    
    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.
    
    This 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.
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 $T$

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

$\forall q \in Q(\tau') - \{q_0\}$ in a *level-by-level order*, do

$\forall p < q$ do

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

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

\[\text{try to restore subsequentiality by } \text{Derived Merging}, \]

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

\[\text{if } \text{"Derived Merging" possible then } \tau = \text{merge}(\tau, q', q'')\]

end while

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

end $\forall p$

end $\forall q$

end OSTIA

---

**An Example of OSTIA state-merging process**

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

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

X={a,b} ; Y={A,B} ; T={(b,B), (a,AB), (bb,BA), (ba,BB), (aa,AAB)}
An Example of OSTIA state-merging process

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

The Onward Subsequential Transducer Inference Algorithm (OSTIA)

INPUT: input-output pairs \( T \subset (X^* \times Y^*) \); OUTPUT: OST \( \tau \) consistent with \( T \)

\[
\tau := \text{OTST}(T) \; ; \; q := \text{first}(\tau) \\
\text{while} \; q < \text{last}(\tau) \; \text{do} \{ \\
\quad q := \text{next}(\tau, q) \; ; \; q' := \text{first}(\tau) \\
\quad \text{while} \; q' < q \; \text{do} \{ \\
\quad \quad \text{if} \; \sigma(q') = \sigma(q) \; \text{or} \; \sigma(q') = \emptyset \; \text{or} \; \sigma(q) = \emptyset \; \text{then} \{ \\
\quad \quad \quad \tau' := \tau \; ; \; \text{merge}(\tau, q, q') \\
\quad \quad \text{while} \; \neg\text{subsequential}(\tau) \; \text{do} \{ \\
\quad \quad \quad \text{let} \; (r, a, v, s), (r, a, v', s') \; \text{be two edges of} \; \tau \; \text{that} \\
\quad \quad \quad \text{violate the} \; \text{subsequential} \; \text{condition, with} \; s' < s; \\
\quad \quad \quad \text{if} \; s' < q \; \text{and} \; v \not\in Pr(v) \; \text{then} \; \text{exitwhile}; \\
\quad \quad \quad u := \text{lcp}(v, v'); \\
\quad \quad \quad \text{push-back}(\tau, u^{-1}v', (r, a, v', s')); \\
\quad \quad \quad \text{push-back}(\tau, u^{-1}v, (r, a, v, s)); \\
\quad \quad \quad \text{if} \; \sigma(s') = \sigma(s) \; \text{or} \; \sigma(s') = \emptyset \; \text{or} \; \sigma(s) = \emptyset \; \text{then} \; \text{merge}(\tau, s, s') \; \text{else} \; \text{exitwhile}; \\
\quad \quad \} \; \text{// while} \; \neg\text{subsequential}(\tau) \\
\quad \text{if} \; \neg\text{subsequential}(\tau) \; \text{then} \; \tau := \tau' \; \text{else} \; \text{exitwhile}; \\
\quad \} \; \text{// if} \; \sigma(q') = \sigma(q) \\
\quad q' := \text{next}(\tau, q'); \\
\} \; \text{// while} \; q' < q \\
\} \; \text{// while} \; q < \text{last}(\tau)
Properties of OSTIA learning

[Oncina, Garcia & 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
  - $n = \sum_{(x,y) \in T} |x|$, (overall length of input strings)
  - $m = \max_{(x,y) \in T} |x|$ (longest output string)
  - $k = |X|$ (size of input alphabet).

$\Rightarrow$ 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]
  - (Subset of) ATIS [Vidal,94]

- **Language Translation:**
  - MLA [Castellanos et al.,94]
  - Traveler Task [Amengual et al., 95-99]
Language Understanding through semantic decoding

Given a speech or text input sentence, produce an output which can be used to drive the actions specified in this sentence.

TYPICAL EXAMPLES:

- ATIS (Air Travel Information Systems):
  - **input**: Spontaneous English Sentences
  - **output**: Formal Query commands to the ATIS Data Base

- BDGEO (Spanish Geographic Quest):
  - **input**: Natural Language Spanish Sentences
  - **output**: Formal Query commands to BDGEO

- MLA (“Miniature Language Acquisition [Feldman et al., 90]):
  - **input**: Quasi-natural English Sentences
  - **output**: First-Order Predicate Logic Formulae

A simple experimental language understanding task: MLA

[Feldman et al., 90]

- Involves description and manipulation of simple visual scenes.
- Originally introduced as a challenging Language Learning task with a fairly simple syntax and small lexicon (about 30 words).
- Extended, as required, to study the impact of increasing complexity, vocabulary size, etc.

Examples:

- a medium light square and a circle are far above a light circle and a medium square
- a large dark triangle is added far to the left of the square and the medium circle
- the large circle which is above the square and the medium triangle is removed
MLA: language understanding through semantic decoding  
[Castellanos et al., 94-98]

- Visual scenes of MLA “understood” in terms of (first-order) logic formulae.
- Objects = Variables: $x, y, z, w$ (allow up to four objects in a scene).
- 8 unary predicates on variables for shape, shade and size
- 9 (0-ary or binary) predicates for object relative positions (above, below, far below, to the right, touch, etc).
- Three increasingly non-monotone representations for object relations: L1, L2, L3. Translation into L1 is purely sequential, subsequential for L2 and L3.

Examples:

*a small triangle* touches *a medium light circle* and *a large square*

L1: \((Sm(x) \land T(x)) \land T(m) \land Li(z) \land C(z) \land La(w) \land S(w)\)
L2: \((Sm(x) \land T(x) \land To(x,z) \land M(z) \land Li(z) \land C(z) \land To(x,w) \land La(w) \land S(w)\)
L3: \((Sm(x) \land T(x) \land M(z) \land Li(z) \land C(z) \land La(w) \land S(w) \land To(x,z) \land To(x,w)\)

Air Travel Information System (ATIS): semantic decoding

Translate English sentences into a semantic representation in terms of “Pseudo English” (PE) formal queries.

Examples:

*show all flights and fares from \(<\text{city}>\) to \(<\star\text{city}>\)*
LIST FLIGHTS FROM <CITY> AND TO <CITY> ALONG WITH FARES

*I’d like information on \(<\text{airline}>\) flight from \(<\text{city}>\) to \(<\star\text{city}>\)*
LIST FLIGHTS FROM <CITY> AND TO <CITY> AND <AIRLINE>

*I’d like to find cheapest fare one-way fare from \(<\text{city}>\) to \(<\star\text{city}>\)*
LIST CHEAPEST ONE-WAY FARES CHARGED FOR FLIGHTS FROM <CITY> AND TO <CITY>

*please tell me about ground transportation from \(<\text{city}>\) airport to downtown \(<\star\text{city}>\)*
LIST GROUND SERVICES PROVIDED FOR <AIRPORT> AND PROVIDED FOR <CITY>

*what airline is \(<\text{airline}>\) abbreviation for*
LIST AIRLINES WHOSE AIRLINE CODE IS <AIRLINE>

English sentences in lowercase, Pseudo-English commands in capitals. Tokens within angular brackets are “generic non-terminals” or bilingual categories.
Semantic decoding: OSTIA learning results

Evolution of test-set semantic error and size of the OSTIA learned transducers for increasing amounts of training data.

MLA-L3 (10k test sentences)
(similar results for L2; slightly better for L1)

ATIS (146 test sentences)
small subset of short, class A sentences)
Machine Translation (MT) and Subsequential Transduction

- Translation between languages can be modeled by Finite State (FS) mappings

- An important advantage of FS Translation Models is their great adequacy to be used for speech-input MT

- Theoretically speaking, most language pairs involve only subsequential mappings (*output text can be produced without having to wait until the end of the input discourse!*)

- In practice, many language pairs do involve only short-term input/output asynchronies

- *Subsequential Transducers* can be appropriate for *Limited Domain MT applications*

A simple experimental Machine Translation task: MTA

[Feldman et al., 90] [Castellanos et al., 94]

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

- Reformulated for Machine Translation and *extended*, as required, to study the impact of increasing degree of input-output *non-monotonicity*, *vocabulary size*, etc.

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 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>WER</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!
A Difficult-to-learn (partial) Subsequential Transduction

Let \( t : \{a, b, c\}^* \rightarrow \{0, 1, 2\}^* \) be a partial Subsequential function defined as:

\[
t = \{(a^m, b^m)|m \geq 0\} \cup \{(ac^m, b^m)|m, n \geq 0\} \cup \{(bc^m|c^{2n+1}, b^m|c^{2n+1}+1)|m, n \geq 0\}
\]

A Subsequential Transducer realizing \( t \):

Samples of \( t \), up to input length 6:

\[
(, ) \quad (cba, 21) \quad (cccc, 2222) \quad (bccccccc, 11111) \\
(a, ) \quad (cca, 22) \quad (accc, 0000) \quad (accccc, 00000) \\
(c, 2) \quad (ccc, 222) \quad (cbccc, 2111) \quad (ccbcccc, 22111) \\
(bc, 1) \quad (bcc, 111) \quad (ccacc, 2200) \quad (cccacc, 22200) \\
(ca, 2) \quad (cacc, 200) \quad (ccbc, 2221) \quad (ccccbc, 22221) \\
(cc, 22) \quad (ccbc, 221) \quad (cccca, 2222) \quad (ccccca, 22222) \\
(acc, 00) \quad (ccca, 222) \quad (cccccc, 222222) \quad (cccccc, 222222)
\]

No transduction example can help distinguish the states \( q \) and \( q' \).
Onward Tree Subsequential Transducer and OSTIA result

OTST of a sample of \( t \) consisting of all the input-output pairs up to an input length of 6.

Transducer yield by OSTIA from this OTST.

Index

1 Subsequential Transduction: “OSTI” Algorithm \( \triangleright \) 2

○ 2 Using input/output syntactic constraints: OSTIA-DR \( \triangleright \) 29

3 OSTIA-DR: improving scalability \( \triangleright \) 45

4 Bibliography \( \triangleright \) 70
Helping OSTIA with input/output syntactic constraints

Two kinds 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

**OSTIA-DR**

[Oncina,93]

- The use of Domain (and Range) information can be accomplished by labeling each state of the initial Onward Tree Subsequential Transducer (OTST) with the name of the state of the Domain (or Range) FS Model that would be reached with the corresponding strings.

- The local compatibility rules then include the condition of disallowing the merging of two states if their labels are distinct.

- **The resulting SSTs only accept input sentences and only produce output sentences compatible with the syntactic constraints represented by the FSMs used**
  
  ▶ This becomes essential for imperfect text or speech input.

- Using OSTIA-DR, the class of partial Subsequential Functions can be identified in the limit.
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:**

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

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

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

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

**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 \( \tau' \)

**end while**

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

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

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

**end** OSTIA

---

**FS input and output models for the “difficult transduction”**

Finite State Domain (input) model

Finite State Range (output) model
OSTIA-D learning

Training set: \( T = \{ (a, \lambda), (ace, 00), (acce, 0000), (bc, 1), (bece, 111), (c, 2) \} \)

OSTIA-R learning

Training set: \( T = \{ (a, \lambda), (ace, 00), (acce, 0000), (bc, 1), (bece, 111), (c, 2), (cc, 22) \} \).
Pattern Recognition Machine Translation

Using input-language constraints: OSTIA-D

INPUT: input-output pairs, \( T \subset (X^* \times Y^*) \), Finite-State model, \( G_D \), of the Domain \( (X^*) \)
OUTPUT: OST \( \tau \) consistent with \( T \) and \( G_D \)

\[ \tau := \text{OTST}(T); \quad q := \text{first}(\tau); \]
\[ \text{while } q < \text{last}(\tau) \{ \]
\[ \quad q := \text{next}(\tau, q); \quad q' := \text{first}(\tau); \]
\[ \quad \text{while } q' < q \{ \]
\[ \quad \quad \text{if } (\sigma(q') = \sigma(q) \text{ or } \sigma(q') = \emptyset \text{ or } \sigma(q) = \emptyset) \text{ and } \]
\[ \quad \quad \quad \delta_D(p_0, \text{input_prefix}(q')) = \delta_D(p_0, \text{input_prefix}(q)) \text{ then } \]
\[ \quad \quad \quad \quad \tau' := \tau; \quad \text{merge}(\tau, q', q); \]
\[ \quad \quad \text{while } \neg \text{subsequential}(\tau) \{ \]
\[ \quad \quad \quad \text{let } (r, a, v, s), (r, a, v', s') \text{ be two edges of } \tau \text{ that } \]
\[ \quad \quad \quad \text{violate the subsequential condition, with } s' < s; \]
\[ \quad \quad \quad \text{if } s' < q \text{ and } v' \not\in Pr(v) \text{ then } \text{exitwhile} \]
\[ \quad \quad \quad u := \text{lc}(v, v'); \]
\[ \quad \quad \quad \text{push_back}(\tau, u^{-1}v, (r, a, v, s')); \quad \text{push_back}(\tau, u^{-1}v, (r, a, v, s)); \]
\[ \quad \quad \quad \text{if } \sigma(s') = \sigma(s) \text{ or } \sigma(s') = \emptyset \text{ or } \sigma(s) = \emptyset \text{ then } \]
\[ \quad \quad \quad \text{merge}(\tau, s', s) \text{ else } \text{exitwhile} \]
\[ \quad \quad \} \] // while \( \neg \text{subsequential}(\tau) \)
\[ \quad \quad \text{if } \neg \text{subsequential}(\tau) \text{ then } \tau := \tau' \text{ else } \text{exitwhile} \]
\[ \quad \} \] // while \( q' < q \)
\[ \} \] // while \( q < \text{last}(\tau) \)


Pattern Recognition Machine Translation

Using output-language constraints: OSTIA-R

INPUT: input-output pairs, \( T \subset (X^* \times Y^*) \), Finite-State model, \( G_R \), of the Range \( (X^*) \)
OUTPUT: OST \( \tau \) consistent with \( T \) and \( G_R \)

\[ \tau := \text{OTST}(T); \quad q := \text{first}(\tau); \]
\[ \text{while } q < \text{last}(\tau) \{ \]
\[ \quad q := \text{next}(\tau, q); \quad q' := \text{first}(\tau); \]
\[ \text{while } q' < q \{ \]
\[ \quad \text{if } (\sigma(q') = \sigma(q) \text{ or } \sigma(q') = \emptyset \text{ or } \sigma(q) = \emptyset) \text{ and } \]
\[ \quad \quad \delta_R(p_0, \text{output_prefix}(q')) = \delta_R(p_0, \text{output_prefix}(q)) \text{ then } \]
\[ \quad \quad \quad \tau' := \tau; \quad \text{merge}(\tau, q', q); \]
\[ \quad \quad \text{while } \neg \text{subsequential}(\tau) \{ \]
\[ \quad \quad \quad \text{let } (r, a, v, s), (r, a, v', s') \text{ be two edges of } \tau \text{ that } \]
\[ \quad \quad \quad \text{violate the subsequential condition, with } s' < s; \]
\[ \quad \quad \quad \text{if } s' < q \text{ and } v' \not\in Pr(v) \text{ then } \text{exitwhile} \]
\[ \quad \quad \quad u := \text{lc}(v, v'); \]
\[ \quad \quad \quad \text{push_back}(\tau, u^{-1}v, (r, a, v, s')); \quad \text{push_back}(\tau, u^{-1}v, (r, a, v, s)); \]
\[ \quad \quad \quad \text{if } \sigma(s') = \sigma(s) \text{ or } \sigma(s') = \emptyset \text{ or } \sigma(s) = \emptyset \text{ then } \]
\[ \quad \quad \quad \text{merge}(\tau, s', s) \text{ else } \text{exitwhile} \]
\[ \quad \} \] // while \( \neg \text{subsequential}(\tau) \)
\[ \quad \text{if } \neg \text{subsequential}(\tau) \text{ then } \tau := \tau' \text{ else } \text{exitwhile} \]
\[ \} \] // while \( q' < q \)
\[ \} \] // while \( q < \text{last}(\tau) \)

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

Translation Word Error Rates for the Extended MTA Feldman's Task, as a function of the Training Set size supplied to OSTIA and OSTIA-DR (with 4-Gram Language Models)

Test Set: 10,000 independent input sentences.

<table>
<thead>
<tr>
<th>Training Set Size</th>
<th>OSTIA WER</th>
<th>States</th>
<th>Edges</th>
<th>OSTIA-DR WER</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>
<td>55.1%</td>
<td>813</td>
<td>2023</td>
</tr>
<tr>
<td>2,000</td>
<td>57.0%</td>
<td>846</td>
<td>3197</td>
<td>47.1%</td>
<td>1406</td>
<td>3353</td>
</tr>
<tr>
<td>4,000</td>
<td>51.8%</td>
<td>1598</td>
<td>5970</td>
<td>30.1%</td>
<td>1686</td>
<td>4051</td>
</tr>
<tr>
<td>8,000</td>
<td>3.4%</td>
<td>186</td>
<td>891</td>
<td>1.4%</td>
<td>244</td>
<td>719</td>
</tr>
<tr>
<td>16,000</td>
<td>0.0%</td>
<td>17</td>
<td>206</td>
<td>0.0%</td>
<td>100</td>
<td>363</td>
</tr>
</tbody>
</table>

Using Input/Output syntactic constraints, translation errors can be reduced by a factor of two.
MTA OSTIA and OSTIA-DR learning: impact of noisy text input and input–output language syntactic constraints

Spanish-English Translation Word Error Rates 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.

Test Set: 10,000 clean and 5%-distorted independent input sentences.

<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

MTA OSTIA and OSTIA-DR Learning: examples of distorted input sentences and the obtained translations

I=Original Input; D=5% Distorted Input; T=System Translation.

Correctly Translated:

I: se elimina el círculo grande y claro que está muy por encima del triángulo oscuro y del cuadrado • mediano
D: se elimina y círculo grande y claro • está muy por encima • triángulo oscuro y del cuadrado un mediano
T: the large light circle which is far above the dark triangle and the medium square is removed

I: un • círculo mediano y claro está debajo de un cuadrado pequeño y claro y un triángulo pequeño y oscuro
D: un tocan círculo mediano y claro • debajo de un cuadrado pequeño claro y se triángulo pequeño y oscuro
T: a medium light circle is below a small light square and a small dark triangle

Translation Errors:

I: se • elimina el círculo que está muy a la izquierda del círculo oscuro y del triángulo mediano y oscuro
D: se de de el • que está muy a la izquierda del círculo oscuro y del triángulo mediano y oscuro
T: the square which is far to the left of the dark circle and the medium dark triangle is removed

I: se añade un triángulo mediano y claro muy a la derecha del cuadrado mediano y oscuro y del círculo pequeño y oscuro
D: se añade un triángulo la y claro muy a la derecha del cuadrado mediano y oscuro oscuro claro círculo pequeño y oscuro
T: a small light triangle is added far to the right of the medium dark square and the small dark circle
3-TSS Automaton (entailing 3-Gram constraints)
OSTIA-DR–learned SST for Spanish-English MTA

(using both Domain and Range 3-Gram constraints)

Index

1 Subsequential Transduction: “OSTI” Algorithm ▷ 2

2 Using input/output syntactic constraints: OSTIA-DR ▷ 29

3  OSTIA-DR: improving scalability ▷ 45

4 Bibliography ▷ 70
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:

(se elimina un triángulo grande y claro, a large light triangle is removed)

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:

$n \Rightarrow$ Bilingual Categorization
[Vilar, Marzal, Vidal, Eurospeech-95]:
While the direct approach degrades rapidly with increasing vocabulary sizes, categorization largely prevents accuracy degradation.

$k \Rightarrow$ Partial Alignment and Word Reordering
[Vilar, Vidal, Amengual, Llorens, ECAI-96, SPECOM-96]:
Training-data requirements can be reduced dramatically.
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 very small MTA Spanish-English training set

se añade un triángulo claro ↔ a light triangle is added
se añade un círculo claro ↔ a light circle is added
se añade un triángulo oscuro ↔ a dark triangle is added
se añade un círculo oscuro ↔ a dark circle is added
se elimina un triángulo claro ↔ a light triangle is removed
se elimina un círculo claro ↔ a light circle is removed
se elimina un triángulo oscuro ↔ a dark triangle is removed
se elimina un círculo oscuro ↔ a dark circle is removed

A Categorized version of this Training Set

se $ACCION un $FORMA $COLOR ↔ a $COLOR $SHAPE is $ACTION
Subsequential Transducer for the very small MTA
Spanish-English training set

Size grows very fast with the number of words in each category.

Categorized Transducer

Size no longer depends on the number of words in each category.

MTA extensions for experimentation with Bilingual Categories

Four extensions to the (extended) Feldman’s MTA task:

- EXT1: 6 shapes, 3 sizes, 2 shades (Voc.: 37/28 Spanish/English words)
- EXT2: 12 shapes, 5 sizes, 4 shades/colors (Voc.: 50/36 words)
- EXT3: 18 shapes, 7 sizes, 6 shades/colors (Voc.: 63/48 words)
- EXT4: 118 shapes, 57 sizes, 56 shades/colors (Voc.: 363/248 words)
MTA: cutting down the impact of increasing vocabulary using Bilingual Categories

[Vilar, Marzal and Vidal, Eurospeech-95]

Translation Sentence Error Rate (in %) for two training-set sizes and increasing vocabulary sizes (3 categories: NOUN, ADJ, ADV).
Test set: 10,000 independent sentences.

<table>
<thead>
<tr>
<th>Voc.Sizes</th>
<th>8,000 Train. Pairs</th>
<th>32,000 Train. Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Direct</td>
<td>Categ.</td>
</tr>
<tr>
<td>37/28</td>
<td>3.1</td>
<td>0.9</td>
</tr>
<tr>
<td>50/38</td>
<td>42.1</td>
<td>1.5</td>
</tr>
<tr>
<td>63/48</td>
<td>62.5</td>
<td>3.0</td>
</tr>
<tr>
<td>363/248</td>
<td>91.3</td>
<td>3.4</td>
</tr>
</tbody>
</table>

While the direct approach degrades rapidly with increasing vocabulary sizes, categorization keeps the accuracy essentially unchanged.

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 12.8 / 7.0

(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.

Despiértenos mañana a las ocho menos cuarto, por favor.
Wake us up tomorrow at a quarter to eight, 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: dates, times-of-day, 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 Expectation-Maximisation and Viterbi re-estimation.

![Graph showing Translation WER (%) vs Different Training Pairs (thousands)](image)

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

Traveler Task: summary of text-input results

Impact of using Categories and Error Correcting Parsing

- OSTIA-DR learned Subsequential Transducers
- Training based on the largest training sets available
- Error model parameters estimated from artificially distorted input text
- Test-set: clean (undistorted) independent input text

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>OSTIA-DR (baseline)</th>
<th>OSTIA-DR + Categories</th>
<th>OSTIA-DR + Categories + ECP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish-English</td>
<td>13.33 %</td>
<td>0.74 %</td>
<td>0.18 %</td>
</tr>
<tr>
<td>Spanish-German</td>
<td>29.86 %</td>
<td>1.23 %</td>
<td>0.54 %</td>
</tr>
<tr>
<td>Spanish-Italian</td>
<td>17.60 %</td>
<td>2.54 %</td>
<td>0.51 %</td>
</tr>
</tbody>
</table>
Traveler Task: human subjective assessment results

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

- Comparison of EuTrans results with translations provided by low-cost commercial translation packages, adapted to the Traveler Task.
- Human subjective results based on three experts.

<table>
<thead>
<tr>
<th></th>
<th>Spanish-to-German</th>
<th>Spanish-to-English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EuTRANS</td>
<td>EuTRANS Power Translator Spanish Translator Assistant</td>
</tr>
<tr>
<td>PCT</td>
<td>81.7%</td>
<td>87.3%</td>
</tr>
<tr>
<td>PCIT</td>
<td>93.3%</td>
<td>90.3%</td>
</tr>
<tr>
<td>UM</td>
<td>+0.86</td>
<td>+0.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.57</td>
</tr>
</tbody>
</table>

- **PCT**: Percentage of correct translations
- **PCIT**: Percentage of correctly intelligible translations
- **UM**: An approximate usefulness measure

### Automatic bilingual word clustering

As task complexity and diversity increase, automated methods are required to discover the bilingual categories which are actually relevant in a given corpus of the task.

A basic idea:

- Modify well-known, monolingual, $K$-means style word clustering techniques, by including translation information.
- Derive this information from an initial bilingual (probabilistic) dictionary.
- This dictionary can be obtained manually and/or using simple statistical techniques such as the IBM-1 translation model.

*Preliminary experiments show that techniques based on this idea often supply very adequate bilingual clusters of (individual) words.*
Cutting down the impact of increasing vocabulary size \( (n) \) and degree of non-monotonicity \( (k) \)

Approaches:

\( n \implies \text{Bilingual Categorization} \)

[Vilar, Marzal, Vidal, Eurospeech-95]:

While the direct approach degrades rapidly with increasing vocabulary sizes, categorization largely prevents accuracy degradation.

\( k \implies \text{Partial Alignment and Word Reordering} \)

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

Training-data requirements can be reduced dramatically.

A small training set from the MTA task

\[
\begin{align*}
\text{se elimina un triángulo grande y claro} & \iff \text{a large light triangle is removed} \\
\text{se elimina un triángulo pequeño y claro} & \iff \text{a small light triangle is removed} \\
\text{se elimina un círculo grande y claro} & \iff \text{a large light circle is removed} \\
\text{se elimina un círculo pequeño y claro} & \iff \text{a small light circle is removed} \\
\text{se elimina un triángulo grande y oscuro} & \iff \text{a large dark triangle is removed} \\
\text{se elimina un triángulo pequeño y oscuro} & \iff \text{a small dark triangle is removed} \\
\text{se elimina un círculo grande y oscuro} & \iff \text{a large dark circle is removed} \\
\text{se elimina un círculo pequeño y oscuro} & \iff \text{a small dark circle is removed} \\
\text{se añade un triángulo grande y claro} & \iff \text{a large light triangle is added} \\
\text{se añade un círculo grande y claro} & \iff \text{a large light circle is added} \\
\text{se añade un círculo pequeño y claro} & \iff \text{a small light circle is added} \\
\text{se añade un triángulo grande y oscuro} & \iff \text{a large dark triangle is added} \\
\text{se añade un círculo pequeño y oscuro} & \iff \text{a small dark circle is added}
\end{align*}
\]
Transducer for the small MTA training set

Size grows exponentially with the number of words to be delayed.

Coping with increasing input/output non-monotonicity

[Vilar et al., 1996]

Words of the (training) output sentences can be easily reordered on the base of partial alignments, which can be obtained, e.g., using a probabilistic bilingual dictionary such as the one obtained by training an IBM-1 translation model.
The Reordering Algorithm

Reordering is done by scanning the output sentence from left to right and creating a new reordered and bracketed sentence along the way.

<table>
<thead>
<tr>
<th>Step</th>
<th>Word</th>
<th>Result (reordered and bracketed sentence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>large</td>
<td>a large</td>
</tr>
<tr>
<td>3</td>
<td>light</td>
<td>a large light</td>
</tr>
<tr>
<td>4</td>
<td>triangle</td>
<td>a triangle &lt; large light &gt;</td>
</tr>
<tr>
<td>5</td>
<td>is</td>
<td>a triangle &lt; large light &gt; is</td>
</tr>
<tr>
<td>6</td>
<td>removed</td>
<td>a removed &lt; triangle &lt; large light &gt; is &gt;</td>
</tr>
</tbody>
</table>

Transducer for the reordered small MTA training set

The number of states no longer grows exponentially

Learning can be achieved with far less training data
Recovering the correct word order

“Un-reordering” can be easily done with the help of the embedded brackets and a stack:

Reordered sentence:

“a removed < triangle < large light > is >”

<table>
<thead>
<tr>
<th>Step</th>
<th>Word</th>
<th>Stack</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>∅</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>removed &lt;</td>
<td>removed</td>
<td>a</td>
</tr>
<tr>
<td>3</td>
<td>triangle &lt;</td>
<td>removed, triangle</td>
<td>a</td>
</tr>
<tr>
<td>4</td>
<td>large</td>
<td>removed, triangle</td>
<td>a large</td>
</tr>
<tr>
<td>5</td>
<td>light</td>
<td>removed, triangle</td>
<td>a large light</td>
</tr>
<tr>
<td>6</td>
<td>&gt;</td>
<td>removed</td>
<td>a large light triangle</td>
</tr>
<tr>
<td>7</td>
<td>is</td>
<td>removed</td>
<td>a large light triangle</td>
</tr>
<tr>
<td>8</td>
<td>&gt;</td>
<td>∅</td>
<td>a large light triangle</td>
</tr>
</tbody>
</table>

Result: “a large light triangle is removed”

Reordering-based training and translation procedures

[Vilar et al., 1996]

Training: Given a training set $S$ of pairs of input/output sentences $(x, y)$, the proposed training approach proceeds 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 $S'$.

Translation: Given a new test input sentence $x$, the trained system produces a translation $y$ through the following simple steps:

1. Using $T$, obtain the translation $y'$ of $x$.
2. “Un-reorder” $y'$ with the help of its embedded brackets to obtain $y$.
Balancing the brackets

Possible problem:
Transducers learned by OSTIA with output-reordered training data may not perfectly generalise a balanced bracketing for new unseen input test sentences. This becomes even more problematic with noisy (or speech) input.

A simple solution:
Limit the depth of the brackets and perform OSTIA-DR learning using an output finite-state “Language Model” that enforces correct bracketing.

\[
\begin{align*}
\Sigma & \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \\
< & \rightarrow \Sigma & \rightarrow \Sigma & \rightarrow \Sigma & \rightarrow \Sigma
\end{align*}
\]

(\(\Sigma\) represents an edge for each word in the output language vocabulary)

The number of states should match the maximum level of embedding allowed.

This can be combined with conventional (e.g., 3-Gram) output Language Models.

### MTA OSTIA-DR/Word-Reordering results

[Vilar, Vidal, Amengual, ECAI-96]

Spanish-English Translation Word Error Rates for the Extended Feldman’s MTA Task, as a function of the Training Set size.

Test Set: 10,000 5%-distorted independent input sentences.

<table>
<thead>
<tr>
<th>Train. size</th>
<th>Direct</th>
<th>Reordered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>44.0%</td>
<td>17.6%</td>
</tr>
<tr>
<td>2,000</td>
<td>37.8%</td>
<td>6.2%</td>
</tr>
<tr>
<td>4,000</td>
<td>25.2%</td>
<td>2.2%</td>
</tr>
<tr>
<td>8,000</td>
<td>2.7%</td>
<td>1.7%</td>
</tr>
<tr>
<td>16,000</td>
<td>1.7%</td>
<td>1.7%</td>
</tr>
</tbody>
</table>

In brackets, model sizes (states/edges).

Reordering can reduce the demand for training data by a factor of four
Index

1 Subsequential Transduction: “OSTI” Algorithm ▷ 2

2 Using input/output syntactic constraints: OSTIA-DR ▷ 29

3 OSTIA-DR: improving scalability ▷ 45

○ 4 Bibliography ▷ 70

Bibliography


Pattern Recognition approaches to Machine Translation

F. Casacuberta and E. Vidal

Pattern Recognition and Human Language Technology Group
Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain

Finite-State Translation Models based on Alignments

Enrique Vidal
evidal@iti.upv.es

January 2005

E. Vidal – ITI-UPV-DSIC

Index

1 Statistical Alignment Models and Finite-State Transducers ► 2
2 Alignment-controlled state merging: OMEGA ► 5
3 Alignments and bilingual segmentation: GIATI ► 12
4 GIATI revisited: pure statistical learning ► 24
5 Bibliography ► 28
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:

$$
\Pr(x \mid y) = \sum_{a \in A(y, x)} \Pr(x, a \mid y)
$$

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: \( a \subseteq \{1, \ldots, I\} \times \{1, \ldots, J\} \), \( I = |x| \), \( J = |y| \)
- 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:

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\text{per} & \text{favore} & \text{vorrei} & \text{una} & \text{camera} & \text{doppia} \\
\end{array}
\]

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

\[
\begin{array}{ccccccc}
a_1 = 0 & a_2 = 3 & a_3 = 3 & a_4 = 4 & a_5 = 6 & a_6 = 5 & a_7 = 2 \\
\end{array}
\]

\[
\begin{array}{ccccccc}
\text{per} & \text{favore} & \text{vorrei} & \text{una} & \text{camera} & \text{doppia} \\
\end{array}
\]

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

Index

1 Statistical Alignment Models and Finite-State Transducers \( \triangleright 2 \)

2 Alignment-controlled state merging: OMEGA \( \triangleright 5 \)

3 Alignments and bilingual segmentation: GIATI \( \triangleright 12 \)

4 GIATI revisited: pure statistical learning \( \triangleright 24 \)

5 Bibliography \( \triangleright 28 \)
Review of OSTIA State-Merging Learning Procedures

- Build an “onward” tree representation of the training data (a tree in which output strings are as close as possible to the root)

- The traversal of the tree goes in a level by level manner, typically by using a lexicographical order of state names.

- Two kinds of State Merging:
  - Merging based on Local Conditions: involve only the two states under consideration. Different Local Conditions lead to different algorithms.
  - Derived merges: once two states are merged, others may also need to be recursively merged (with the help of possible output substring “Pushing-back”) in order to preserve determinism.

- If a cascade of derived merges fails preserving determinism, the original and all the derived merges are discarded

Local Conditions for State Merging

- 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, 98]: also takes into account alignments and word to word dictionaries.
The Problem of Premature Output

Assume the following situation in the process of OSTIA learning:

OSTIA would join states C and D, yielding:

This entails a bad generalisation (un cuadrado grande, a square), and moreover now A and F could not be joined. This problem can be solved with a new extension to OSTIA called "OMEGA".\(^1\)

\(^1\)For the Spanish "OSTIA Modificado Empleando Garantías y Alineamientos" [Vilar, 98].

State Labeling with the help of a Dictionary and/or Alignment

Suposse that a known dictionary or alignment hints that the translation of grande, pequeño and circulo should be large, small and circle, respectively. This can be used for state labelling as follows:

Now, states C and D cannot be merged, but A and F can, finally yielding:
The OMEGA extension to OSTIA

[Vilar, 1998]

- The initial tree is built taking alignments and/or dictionaries into account to avoid premature output. Each state \( p \) is labelled 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</th>
<th>OMEGA-DR</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 Statistical Alignment Models and Finite-State Transducers ▶ 2

2 Alignment-controlled state merging: OMEGA ▶ 5

3 Alignments and bilingual segmentation: GIATI ▶ 12

4 GIATI revisited: pure statistical learning ▶ 24

5 Bibliography ▶ 28

Regular Grammars and finite state transducers: a morphism theorem

Theorem [Berstel 1979]:

\[ T \subseteq X^* \times Y^* \] is a rational translation if and only if there exist an alphabet \( Z \), a regular language \( L \subseteq Z^* \) and two morphisms \( h_X : Z^* \rightarrow X^* \) and \( h_Y : Z^* \rightarrow Y^* \) 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: general training procedure**

\[
\begin{align*}
P & \subset \Sigma^* \times \Delta^* \quad \text{A sample of training pairs} \\
S & \subset \Gamma^* \quad \text{Corresponding extended strings} \\
E & : P \subset T(E) \quad \text{A finite-state transducer} \\
G & : S \subset L(G) \quad \text{A regular grammar}
\end{align*}
\]

**Learning approach:**

1. Build a labelled corpus (extended symbols) using statistical alignments.

2. Infer a (stochastic) regular grammars using the labelled corpus.

3. Transform the extended symbols of transitions into input/output symbols.
GIATI: First step (Example)

**Using Statistical Alignments to Convert Training Pairs into Training Strings**

**Training pairs:**

- *una camera doppia* → *a double room*
- *una camera* → *a room*
- *la camera singola* → *the single room*
- *la camera* → *the room*

**Aligned sentences:**

<table>
<thead>
<tr>
<th>una camera doppia</th>
<th>una camera</th>
<th>la camera singola</th>
<th>la camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (1) double (3) room (2)</td>
<td>a (1) room (2)</td>
<td>the (1) single (3) room (2)</td>
<td>the (1) room (2)</td>
</tr>
</tbody>
</table>

E. Vidal – ITI-UPV-DSIC January 2005 Page 7.16

**GIATI: First step: the labelling procedure**

Let $x, y$ and $a$ be an input string, an output string and an alignment function, respectively, $z$ is the labelled string with $|z| = |x|$ and:

For $1 \leq i \leq |z|$

$$z_i = \begin{cases} x_i + y_j + y_{j+1} + \ldots + y_{j+l} & \text{if } \exists j : a(j) = i \text{ and } \exists j' < j : a(j') > a(j) \\
 x_i & \text{otherwise} \end{cases}$$

and for $j'' : j \leq j'' \leq j + l, a(j'') \leq a(j)$

**Aligned training pairs:**

- *una camera doppia* → *a (1) double (3) room (2) → una+a camera doppia+double+room*
- *una camera* → *a (1) room (2) → una+a camera+room*
- *la camera singola* → *the (1) single (3) room (2) → la+the camera singola+single+room*
- *la camera* → *the (1) room (2) → la+the camera+room*
**GIATI: Second step**

**FROM TRAINING STRINGS TO GRAMMARS: N-GRAMS**

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

**PROBLEM:** Non-seen events in the training set.

**COMMON SOLUTION:** Smoothing.

---

**GIATI: Third step**

**FROM GRAMMARS TO TRANSDUCERS: INVERSE LABELLING**

<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>

---
GIATI results

With IBM Model 5 alignments and back-off smoothed \( n \)-grams, for the standard corpus EUTRANS-0
(171,481 different training pairs, Vocabularies: 689/514 words)

<table>
<thead>
<tr>
<th>( n )</th>
<th>states</th>
<th>transitions</th>
<th>WER (%)</th>
<th>SER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4,056</td>
<td>67,235</td>
<td>8.8</td>
<td>50.1</td>
</tr>
<tr>
<td>3</td>
<td>33,619</td>
<td>173,500</td>
<td>4.7</td>
<td>27.2</td>
</tr>
<tr>
<td>4</td>
<td>110,321</td>
<td>364,373</td>
<td>4.1</td>
<td>23.2</td>
</tr>
<tr>
<td>5</td>
<td>147,790</td>
<td>492,840</td>
<td>3.8</td>
<td>20.5</td>
</tr>
<tr>
<td>6</td>
<td>201,319</td>
<td>663,447</td>
<td>3.6</td>
<td>19.0</td>
</tr>
<tr>
<td>7</td>
<td>264,868</td>
<td>857,275</td>
<td>3.4</td>
<td>18.0</td>
</tr>
<tr>
<td>8</td>
<td>331,598</td>
<td>1,050,949</td>
<td>3.3</td>
<td>17.4</td>
</tr>
<tr>
<td>9</td>
<td>391,812</td>
<td>1,218,367</td>
<td>3.3</td>
<td>17.2</td>
</tr>
<tr>
<td>10</td>
<td>438,802</td>
<td>1,345,278</td>
<td>3.2</td>
<td>16.8</td>
</tr>
<tr>
<td>11</td>
<td>471,733</td>
<td>1,432,954</td>
<td>3.1</td>
<td>16.4</td>
</tr>
<tr>
<td>12</td>
<td>492,620</td>
<td>1,485,370</td>
<td>3.1</td>
<td>16.4</td>
</tr>
</tbody>
</table>

Comparative experiments: benchmark corpora

**EUTRANS-I CORPUS [VIDAL 1997]**

<table>
<thead>
<tr>
<th>Train: Sentences</th>
<th>Spanish</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>97,131</td>
<td>99,292</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>686</td>
<td>513</td>
</tr>
</tbody>
</table>

Test: Sentences 2,996
| Words          | 35,023  | 35,590  |
| Bigram Perplexity | 8.6     | 5.2     |

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>Train: Sentences</th>
<th>Italian</th>
<th>English</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

Test: Sentences 300
| Words          | 6,121   | 7,243   |
| Bigram Perplexity | 31      | 25      |

Transcriptions of Italian-English spontaneous sentences, person-to-person communication in the hotel framework.
OSTIA / OMEGA / GIATI comparative results

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

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Method</th>
<th>Assisted by</th>
<th>n-grams</th>
<th>WER</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>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>

\[ECP = \text{Error-Correcting Parsing}
BOS = \text{Back-Off Smoothing}
ABS = \text{Automatic Bilingual Segmentation}
IBM^k = \text{IBM Model } k \text{ statistical alignments}
IBM2' = \text{Symetrized IBM2}\]

Summary of Stochastic Finite-State MT 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>Languages</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>Vocabularies</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>Training (words)</td>
<td>110K</td>
<td>4.5M</td>
<td>100K</td>
<td>50K</td>
<td>600K</td>
<td>90K</td>
<td>6M</td>
</tr>
</tbody>
</table>

<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>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>
</tbody>
</table>

<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>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, Basc
PB = Phrase-based alignment models
AT = Alignment Templates
Index

1 Statistical Alignment Models and Finite-State Transducers ▷ 2
2 Alignment-controlled state merging: OMEGA ▷ 5
3 Alignments and bilingual segmentation: GIATI ▷ 12
  ○ 4 GIATI revisited: pure statistical learning ▷ 24
5 Bibliography ▷ 28

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,

  $\mu : \{1, \ldots, J\} \rightarrow \{1, \ldots, I\}$ 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_k^K} \prod_{k=1}^J Pr(x_k, y_{\mu_{k-1}+1}^{\mu_k}, y_{\mu_{k-1}+1}^{\mu_k-1}) \cdot Pr(\$$, $$|x_{J-n+2}, 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 \((x_{k-n+1}^{k-1}, y_{k-n+1}^{\mu_k-1})\) in the training set;
  - The probability of a transition between two states \((x_{k-n+1}^{k-1}, y_{k-n+1}^{\mu_k-1})\) and \((x_{k-n+1}^{k-1}, y_{k-n+1}^{\mu_k-1})\) is \(Pr(x_k, y_{k-1}^{\mu_k} | x_{k-n+1}^{k-1}, y_{k-n+1}^{\mu_k-1})\) with \(x_k\) as source symbol and \(y_{k-n+1}^{\mu_k-1}\) as the target string;
  - The probability that \((x_{k-n+1}^{k-1}, y_{k-n+1}^{\mu_k-1})\) of a final state is \(Pr(\$, \$ | x_{k-n+1}^{k-1}, y_{k-n+1}^{\mu_k-1})\).
- **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
Index

1 Statistical Alignment Models and Finite-State Transducers ▷ 2

2 Alignment-controlled state merging: OMEGA ▷ 5

3 Alignments and bilingual segmentation: GIATI ▷ 12

4 GIATI revisited: pure statistical learning ▷ 24
   ○ 5 Bibliography ▷ 28

Bibliography


Pattern Recognition Approaches to Machine Translation
E. Vidal and F. Casacuberta
Pattern Recognition and Human Language Technology Group
Departament de Sistemes Informàtics i Computació
Institut Tecnològic d’Informàtica
Universitat Politècnica de València

8: Recursive Alignment Models

Francisco Casacuberta Nolla
fcn@iti.upv.es

24-28 January 2005

Index

1 Introduction ▷ 2
2 A recursive alignment model: MAR ▷ 11
3 Stochastic inversion transduction grammar ▷ 30
4 Bilingual Recursive Alignments ▷ 34
5 Bibliography ▷ 47
Exemple of word alignments

taxi . . . . . ■ . . .
un . . . . . ■ . . .
pídame ■ ■ ■ ■ . . . .
, . . . . . ■ . .
favor . . . . . . ■
por . . . . . . ■ .
could you ask for a taxi, please?

Exemple of word alignments

<table>
<thead>
<tr>
<th>taxi</th>
<th>un</th>
<th>pidiame</th>
<th>favor</th>
<th>por</th>
<th>could</th>
<th>ask</th>
<th>for</th>
<th>a</th>
<th>taxi</th>
<th>please</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

INVERSION

DIRECT
Exemple of word alignments

por favor, pídame un taxi # could you ask for a taxi, please?

- por favor, #, please?
- pidame un taxi # could you ask for a taxi
- un taxi # a taxi
- taxi # taxi
**Exemple of word alignments**

H. Ney, *Statistical Natural Language Processing*, 2003: Canadian Hansards

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>proposal</td>
<td>proposition</td>
</tr>
<tr>
<td>new</td>
<td>■</td>
</tr>
<tr>
<td>under</td>
<td>■</td>
</tr>
<tr>
<td>fees</td>
<td>■</td>
</tr>
<tr>
<td>collecting</td>
<td>■ ■</td>
</tr>
<tr>
<td>and</td>
<td>■</td>
</tr>
<tr>
<td>administering</td>
<td>■</td>
</tr>
<tr>
<td>cost</td>
<td>■</td>
</tr>
<tr>
<td>anticipated</td>
<td>■</td>
</tr>
<tr>
<td>the</td>
<td>■</td>
</tr>
<tr>
<td>is</td>
<td>■</td>
</tr>
<tr>
<td>What</td>
<td>■</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>En</th>
<th>vertu</th>
<th>les</th>
<th>nouvelles</th>
<th>propositions</th>
<th>quel</th>
<th>est</th>
<th>le</th>
<th>cout</th>
<th>de</th>
<th>perception</th>
<th>les</th>
<th>droits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemoa</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td></td>
<td>■</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>.</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
<tr>
<td>de</td>
<td>.</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
<tr>
<td>marzo</td>
<td>.</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
<tr>
<td>de</td>
<td>.</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>.</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>.</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
<tr>
<td>Lemoan</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
<tr>
<td>En</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>■</td>
<td></td>
<td>■</td>
<td>■■</td>
<td></td>
</tr>
</tbody>
</table>

AMETRA corpus

1996 · ■ ·
de · ■ ·
marzo · ■ ·
de · ■ ·
20 · ■ ·
a · ■ ·
Lemoa · ■ ·
En · ■ ·

F. Casacuberta – DSIC-ITI-UPV 24-28 January 2005
Exemple of word alignments

METEO corpus

sud . . . . . . ■
meitat . . . . . ■ .
seva . . . . . ■ .
la . . . . . . .
en . . . . ■ . .
Llevant . . . ■ . . .
de . . ■ . . . . .
des . . ■ . . . . .
sobretot ■ ■ . . . . .

Index

1 Introduction ▷ 2

2 A recursive alignment model: MAR ▷ 11

3 Stochastic inversion transduction grammar ▷ 30

4 Bilingual Recursive Alignments ▷ 34

5 Bibliography ▷ 47
A Recursive Alignment Model: MAR

\[ MAR = (\text{Recursive Alignment Model})^{-1} \]


*The slides on MAR are modified versions of some material supplied by J.M. Vilar.

- Accounts for differences in word order between languages.
- Assumes hierarchical structured alignments.
- The alignments obtained are particularly adequate to be used in combination with finite-state techniques:
  - Allow to use automatically obtained short phrases (rather than words) for:
    - Bilingual clustering.
    - Reordering of source-target pairs.
MAR’s generative process

The translation of a sentence segment can be decomposed in:

1. Decide whether MAR or IBM has to be used. If IBM is chosen, the segment is translated by it and the process ends.

2. If MAR is used, the sentence is divided in two segments.

3. Each segment is recursively translated (hence the name: Recursive Alignment Model).

4. The resulting translations are concatenated in the original or in the inverse order.

A simple example

Leaves are labelled by $J = |x|$, $I = |y|$, while internal nodes are labelled by $J, b, d : 1 \leq b \leq J = |x|$, $d \in \{D, I\}$, where $b$ is the cut-point of $x$ and $D, I$ indicate that the source-target segments are “Directly” or “Inversely” aligned, respectively.
Formal derivation

First, approximate the translation probability by

\[ \Pr(y|x) \approx P_M(y|x) \]

\[ = \Pr(M = \text{IBM}|x) \cdot P_{M1}(y|x) \]

\[ + \Pr(M = \text{MAR}|x) \cdot P_{MAR}(y|x) \]

\( P_{M1}(y|x) \) is given by IBM-1; \( P_{MAR}(y|x) \) can be written as:

\[ P_{MAR}(y|x) = \sum_{b=1}^{J-1} \Pr(b|x) \cdot \sum_{d \in \{D,R\}} \Pr(d\mid b, x) \cdot \sum_{c=1}^{l-1} \Pr(y_c^d\mid b, d, x) \Pr(y_c^{I+1}\mid b, d, x, y_c^1) \]

Simplifications

1. The choice of the model depends only on the length of the source sentence:

\[ \Pr(M = \text{IBM}|x) \approx M_1(J) \quad \Pr(M = \text{MAR}|x) \approx M_M(J) \]

2. The place for the boundary depends only on the two words adjacent to it:

\[ \Pr(b|x) \approx \frac{B(x_b, x_{b+1})}{\sum_{i=1}^{J-1} B(x_i, x_{i+1})} \]

3. The direction of the concatenation depends on these two words:

\[ \Pr(d = D|b, x) \approx D_D(x_b, x_{b+1}) \quad \Pr(d = R|b, x) \approx D_R(x_b, x_{b+1}) \]

4. The translations of the two halves are independent:

\[ \Pr(y_I^c\mid b, d, x) \approx \begin{cases} P_M(y_I^c\mid x_b^b) & \text{if } d = D \\ P_M(y_I^c\mid x_{b+1}^J) & \text{if } d = R \end{cases} \]

\[ \Pr(y_I^{c+1}\mid b, d, x, y_I^1) \approx \begin{cases} P_M(y_I^{c+1}\mid x_{b+1}^J) & \text{if } d = D \\ P_M(y_I^{c+1}\mid x_1^1) & \text{if } d = R \end{cases} \]
Final form of MAR


\[
Pr(y \mid x) \approx P_M(y \mid x) \\
= M_1(J) \cdot P_{M1}(y \mid x) \\
+ M_M(J) \sum_{b=1}^{J-1} \sum_{i=1}^{J-1} B(x_b, x_{b+1}) \frac{B(x_i, x_{i+1})}{B(x_b, x_{b+1})} \\
\cdot \left( D_D(x_b, x_{b+1}) \sum_{c=1}^{I-1} p_T(y^c_1 \mid x^b_1) \cdot P_M(y^c_{c+1} \mid x^b_{b+1}) \right. \\
+ D_I(x_b, x_{b+1}) \sum_{c=1}^{I-1} p_T(y^I_1 \mid x^b_1) \cdot P_M(y^I_{c+1} \mid x^b_{b+1}) \right)
\]

where \( P_{M1}(y \mid x) \) corresponds to IBM-1 model.

Parameter estimation

*Maximum Likelihood* criterion: Given a sample of example pairs, \( A \), find model parameter values such that the likelihood of \( A \) is maximum. That is, find the maximum of:

\[
\mathcal{L}_A = \prod_{(x,y) \in A} P_M(y \mid x)
\]

This can be (locally optimally) solved through *Expectation Maximization*. Baum Eagon’s inequality is used to estimate all the parameters, except for the \( B \)s which are reestimated using Gopalakrishnan’s.
Parameter estimation: “polynomial” and “rational parameters”

Let \( p \) be a parameter such that \( L \) is polynomial with \( p \) and let \( \mathcal{F}(p) \) be all the other parameters related with \( p \) (i.e.: \( \sum_{q \in \mathcal{F}(p)} q = 1 \)). A reestimated value of \( p \), \( T(p) \) ([Baum & Eagon, 1968]):

\[
T(p) = \frac{p \sum_{q \in \mathcal{F}(p)} \frac{\partial L}{\partial p}}{p \sum_{q \in \mathcal{F}(p)} \frac{\partial L}{\partial q}} = \frac{p \sum_{(x, y) \in A} \left( P_M(y \mid x) \right)^{-1} \frac{\partial P_M(y \mid x)}{\partial p}}{p \sum_{(x, y) \in A} \left( P_M(y \mid x) \right)^{-1} \frac{\partial P_M(y \mid x)}{\partial q}}
\]

Let \( p \) be a parameter such that \( L \) is rational with \( p \) and let \( \mathcal{F}(p) \) be all the other parameters related with \( p \) (i.e.: \( \sum_{q \in \mathcal{F}(p)} q = 1 \)). A reestimated value of \( p \), \( T(p) \) ([Gopalakrishnan et al, 1991]):

\[
T(p) = \frac{\frac{\partial L}{\partial p} + C}{\sum_{q \in \mathcal{F}(p)} \frac{\partial L}{\partial q} + C} = \frac{p \sum_{(x, y) \in A} \left( P_M(y \mid x) \right)^{-1} \frac{\partial P_M(y \mid x)}{\partial p} + C}{p \sum_{(x, y) \in A} \left( P_M(y \mid x) \right)^{-1} \frac{\partial P_M(y \mid x)}{\partial q} + \sum_{q \in \mathcal{F}(p)} C}
\]

Derivative of MAR probabilities

- Let \( m \) be a fixed source sentence length. \( \mathcal{M}_I(m) \) provides the probability of choosing the IBM-1 model, given \( m: \partial P_M(y \mid x) / \partial \mathcal{M}_I(m) \)
- Let \( m \) be a fixed source sentence length. \( \mathcal{M}_M(m) \) provides the probability of choosing the MAR model, given \( m: \partial P_M(y \mid x) / \partial \mathcal{M}_M(m) \)
- Let \( x, x' \) be two fixed source words. \( B(x, x') \) accounts for the probability of placing a boundary point between \( x \) and \( x' \): \( \partial P_M(y \mid x) / \partial B(x, x') \)
- Let \( x, x' \) be two fixed source words. \( \mathcal{D}_D(x, x') \) provides the probability of choosing a Direct alignment: \( \partial p_T(y \mid x) / \partial \mathcal{D}_D(x, x') \)
- Let \( x, x' \) be two fixed source words. \( \mathcal{D}_R(x, x') \) provides the probability of choosing an Inverse alignment: \( \partial p_T(y \mid x) / \partial \mathcal{D}_R(x, x') \)
- Let \( m, n \) be fixed lengths of source/target sentences. \( n(m \mid n) \) accounts for the length distribution of IBM-1 model: \( \partial P_M(y \mid x) / \partial n(m \mid n) \)
- Let \( x, y \) be fixed source/target words. \( l(y \mid x) \) determines the probability that \( y \) be a translation of \( x \): \( \partial P_M(y \mid x) / \partial l(y \mid x) \)
Parameter estimation: Details and simplifications

- 10 Expectation Maximization iterations with neutral initialization.
- The value of $M_I(l)$ set to 0 for $l > 4$.
- The value of $n(l, m)$ is not estimated for $l$ or $m$ greater than four.
- The values of $l(x|y)$ are not estimated for pairs with $l(x|y) < 10^{-5}$.

The resulting estimation algorithm has polynomial time complexity, though it still is very computationally intensive.

Results: Training perplexity evolution

Training set perplexity computed as:

$$PP = m \prod_{(x,y) \in S} (P_M(y|x))^{-1}, \quad m = \sum_{(x,y) \in S} I.$$  

<table>
<thead>
<tr>
<th>Iteration</th>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>506.05</td>
<td>558.24</td>
</tr>
<tr>
<td>1</td>
<td>23.10</td>
<td>23.61</td>
</tr>
<tr>
<td>2</td>
<td>8.78</td>
<td>10.61</td>
</tr>
<tr>
<td>3</td>
<td>5.07</td>
<td>6.26</td>
</tr>
<tr>
<td>4</td>
<td>3.98</td>
<td>4.80</td>
</tr>
<tr>
<td>5</td>
<td>3.45</td>
<td>4.12</td>
</tr>
<tr>
<td>6</td>
<td>3.19</td>
<td>3.74</td>
</tr>
<tr>
<td>7</td>
<td>3.08</td>
<td>3.51</td>
</tr>
<tr>
<td>8</td>
<td>3.01</td>
<td>3.38</td>
</tr>
</tbody>
</table>

Convergence is achieved after a moderate number of iterations.
Spanish-English MAR alignment

\[
\text{deseo reservar dos habitaciones para tres días.}
\]
I want to book two rooms for three days.

\[
\text{deseo reservar dos habitaciones}
\]
I want to book two rooms

\[
\text{deseo}
\]
I want

\[
\text{reservar}
\]
to book

\[
\text{dos habitaciones}
\]
two rooms

\[
\text{dos}
\]
two

\[
\text{habitaciones}
\]
rooms

\[
\text{para tres días.}
\]
for three days.

\[
\text{para tres}
\]
for three

\[
\text{tres}
\]
three

\[
\text{días.}
\]
days.

\[
\text{para}
\]
for

\[
\text{tres}
\]
three

\[
\text{días.
\]
days.

\[
\text{for}
\]
for

\[
\text{esta noche.}
\]
for tonight.

\[
\text{una habitación con televisión para esta noche.}
\]
I want a room with a tv for tonight.

\[
\text{una habitación con}
\]
I want a room with

\[
\text{una}
\]
a

\[
\text{un}
\]
a

\[
\text{habitación con}
\]
room with

\[
\text{televisión para esta noche.}
\]
a tv for tonight.

\[
\text{televisión para}
\]
a tv for

\[
\text{para}
\]
for

\[
\text{esta noche.}
\]
for tonight.

\[
\text{esta noche}
\]
for tonight

\[
\text{tonight}
\]
for tonight
Spanish-English MAR alignment

¿podríamos pagar el recibo con cheques de viaje?
could we pay the bill by traveler check?

¿podríamos pagar el recibo con cheques de viaje?
could we pay the

¿podríamos
could we

¿podríamos
we

¿podríamos pagar el pay the

¿podríamos pagar el
pago

¿podríamos
pago

¿podríamos
el

¿podríamos pagar el
pago

¿podríamos
el

¿podríamos pagar el
pago

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podríamos
el

¿podría...
Spanish-German MAR alignment

A simplified recursive alignment model

H. Ney, *Statistical Natural Language Processing*, STC Doctorate Program, UPC. 2003

\[
\Pr(y, x) \approx \sum_{i, j} \left( \alpha \cdot \Pr(y_1^i, x_1^j) \cdot \Pr(y_{i+1}^f, x_{j+1}^f) + (1 - \alpha) \cdot \Pr(y_1^i, x_{j+1}^f) \cdot \Pr(y_{i+1}^f, x_1^j) \right)
\]

Searching using a bigram target language model and a maximum approach:

\[
Q(j, j', y, y') = \max_{y'', y'''} \left( p_2(y''') | y'' \right) \cdot \max_j \left( \alpha \cdot Q(j, j'', y, y'') \cdot Q(j'' + 1, j', y'', y'), (1 - \alpha) \cdot Q(j, j'', y'', y') \cdot Q(j'' + 1, j', y, y'') \right)
\]
Stochastic inversion transduction grammars


A context-free based approach to bilingual segmentation

For a non-terminal symbol $A$, $B$ and $C$ and for any source word $s$ and any target word $t$,

\[
\begin{align*}
A & \rightarrow \langle B, C \rangle \\
A & \rightarrow [B, C] \\
A & \rightarrow x/y \\
A & \rightarrow x/\lambda \\
A & \rightarrow \lambda/y
\end{align*}
\]
An example

\[
S \rightarrow [A, B] \\
A \rightarrow x_1/y_1 \\
B \rightarrow \langle C, D \rangle \\
C \rightarrow x_2/y_2 \\
D \rightarrow x_3/y_3
\]

\[
x_1 \times x_3 \implies y_1 \times y_3 \times y_2
\]

Stochastic inversion transduction grammars

- Learning:
  - Adapted context-free grammatical inference
  - Inside-outside estimation

- Translation:
  - Adapted Cooker-Younger-Kasami parser algorithm
  - Inside or outside algorithms
Recursive Bilingual Alignments

An alignment between phrases of a source sentence and phrases of a target sentence.

- It represents the translation relations between two sentences.
- It also includes information about the possible reorderings needed in order to generate the target sentence from the source sentence.
- Representation: binary tree.
  - The inner nodes store the reordering directions.
  - The leaf nodes store the translation relations.

*The slides on RBA are modified versions of some material supplied by F. Nevado.*
An example

Prestakuntza didaktoa eta teknikoa informatika irakasgaia ikastoletan garatzeko
Formación didáctica y técnica para desarrollar la asignatura de informática en las ikastolas
An example

Prestakuntza didaktikoa eta teknikoa informatika irakasgaia ikastoletan garatzeko
Formación didáctica y técnica para desarrollar la asignatura de informática en las ikastolas

DIRECT

Prestakuntza didaktikoa eta teknikoa | Formacion didactica y tecnica

INVERSE

garatzeko / para desarrollar

informatika irakasgaia ikastoletan / la asignatura de informatica en las ikastolas

F. Casacuberta – DSIC-ITI-UPV 24-28 January 2005 8: 38
Greedy bilingual recursive alignment

\[ \Pr(y, x) \approx \max_{i,j} \left( \alpha \cdot \Pr(y^1_i, x^1_j) \cdot \Pr(y^1_{i+1}, x^1_{j+1}) + (1 - \alpha) \cdot \Pr(y^1_i, x^1_{j+1}) \cdot \Pr(y^1_{i+1}, x^1_1) \right) \]

- \( \alpha = 0.5 \)
- \( \Pr(y^1_i, x^1_j) \approx P_{M1}(y^1_i, x^1_j) \)

\( (i, j) = \arg\max_{i,j} \left\{ \max \left( P_{M1}(y^1_1, x^1_i), P_{M1}(y^1_{i+1}, x^1_{j+1}), P_{M1}(y^1_i, x^1_{j+1}), P_{M1}(y^1_{i+1}, x^1_1) \right) \right\} \)

Recalign algorithm

A greedy algorithm to compute recursive alignments from a bilingual corpus.

Probability of translating a source phrase into a target phrase \( \rightarrow \) Model 1.

Algorithm:

1. Given \( x \) and \( y \), it computes the most probable breakpoint in each sentence using Model 1.
2. If the translation probability for \( x \) and \( y \) is higher than the translation probability of dividing them:
   - It creates a leaf node where the output sequence is considered to be the translation of the input sequence.
   
   Else:
   - It creates a new inner node of the tree.
   - Apply recursively the algorithm to the left and the right children.
Recalign: variants

- To control the medium length of the generated segments
  \[\Rightarrow\] Combine the translation probabilities with a distribution over the sequences length: LEN modification.

- Model 1 can obtain imprecise divisions
  \[\Rightarrow\] Only allow divisions that are compatible with a word alignment: ALI restriction.
  - Source-to-target (Target-to-source).
  - Symmetrization: union, intersection, refined.

Corpus description

<table>
<thead>
<tr>
<th></th>
<th>EUTRANS-I English-Spanish</th>
<th>DFB Basque-Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>12,960</td>
<td>284,842</td>
</tr>
<tr>
<td>Spanish</td>
<td>134,435</td>
<td>4,203,117</td>
</tr>
<tr>
<td>Sentences</td>
<td>131,707</td>
<td>5,661,564</td>
</tr>
<tr>
<td>Words</td>
<td>514</td>
<td>144,670</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>688</td>
<td>62,412</td>
</tr>
<tr>
<td><strong>Test:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>487</td>
<td>481</td>
</tr>
<tr>
<td>Spanish</td>
<td>491</td>
<td>609</td>
</tr>
<tr>
<td>Sentences</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>Words</td>
<td>126</td>
<td>342</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>149</td>
<td>311</td>
</tr>
<tr>
<td>Trigram Perplexity</td>
<td>3.6</td>
<td>4.6</td>
</tr>
</tbody>
</table>
## Assessment

Given a segmentation $S$ produced by a system and given a reference segmentation $S_r$ produced by an expert,

- **Recall**: Number of bilingual segments that are correct with respect to the number of references:

$$\text{Recall} = \frac{S \cup S_r}{S_r}$$

- **Precision**: Number of bilingual segments that are correct with respect to the number of bilingual segments supplied by the system:

$$\text{Precision} = \frac{S \cup S_r}{S}$$

- **F-measure**:  

$$F - \text{measure} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

## Results: Eutrans-I (Spanish-to-English)

<table>
<thead>
<tr>
<th>Bilingual segmentation</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIATI-labelling</td>
<td>39.22</td>
<td>87.96</td>
<td>54.25</td>
</tr>
<tr>
<td><strong>Recalign</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recalign</td>
<td>37.23</td>
<td>87.10</td>
<td>52.16</td>
</tr>
<tr>
<td>Recalign + LEN</td>
<td>76.86</td>
<td>72.00</td>
<td>74.35</td>
</tr>
<tr>
<td>Recalign + ALI(S-E)</td>
<td>40.01</td>
<td>87.12</td>
<td>54.84</td>
</tr>
<tr>
<td>Recalign + ALI(S-E) + LEN</td>
<td>77.38</td>
<td>71.78</td>
<td>74.48</td>
</tr>
<tr>
<td>Recalign + ALI(∪)</td>
<td>52.21</td>
<td>82.41</td>
<td>63.92</td>
</tr>
<tr>
<td>Recalign + ALI(∪) + LEN</td>
<td>81.63</td>
<td>67.52</td>
<td>73.91</td>
</tr>
<tr>
<td>Recalign + ALI(∩)</td>
<td>38.20</td>
<td>86.57</td>
<td>53.01</td>
</tr>
<tr>
<td>Recalign + ALI(∩) + LEN</td>
<td>76.86</td>
<td>72.00</td>
<td>74.35</td>
</tr>
<tr>
<td>Recalign + ALI(ref)</td>
<td>49.14</td>
<td>84.28</td>
<td>62.08</td>
</tr>
<tr>
<td>Recalign + ALI(ref) + LEN</td>
<td>81.17</td>
<td>68.37</td>
<td>74.22</td>
</tr>
</tbody>
</table>
Pattern Recognition approaches to Machine Translation

Recursive Alignment Models

Results: DFB (Spanish-to-Basque)

<table>
<thead>
<tr>
<th>Bilingual segmentation</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIATI-labelling</td>
<td>63.16</td>
<td>39.13</td>
<td>48.32</td>
</tr>
<tr>
<td>Recalign</td>
<td>75.00</td>
<td>24.21</td>
<td>36.61</td>
</tr>
<tr>
<td>Recalign + LEN</td>
<td>65.03</td>
<td>36.46</td>
<td>46.73</td>
</tr>
<tr>
<td>Recalign + ALI(S-B)</td>
<td>78.26</td>
<td>24.08</td>
<td>36.83</td>
</tr>
<tr>
<td>Recalign + ALI(S-B) + LEN</td>
<td>79.07</td>
<td>24.66</td>
<td>37.60</td>
</tr>
<tr>
<td>Recalign + ALI(∪)</td>
<td>92.16</td>
<td>14.69</td>
<td>25.34</td>
</tr>
<tr>
<td>Recalign + ALI(∪) + LEN</td>
<td>93.37</td>
<td>14.88</td>
<td>25.66</td>
</tr>
<tr>
<td>Recalign + ALI(∩)</td>
<td>76.77</td>
<td>24.29</td>
<td>36.91</td>
</tr>
<tr>
<td>Recalign + ALI(∩) + LEN</td>
<td>74.09</td>
<td>34.01</td>
<td>46.62</td>
</tr>
<tr>
<td>Recalign + ALI(ref)</td>
<td>84.54</td>
<td>21.21</td>
<td>33.91</td>
</tr>
<tr>
<td>Recalign + ALI(ref) + LEN</td>
<td>83.49</td>
<td>30.58</td>
<td>44.76</td>
</tr>
</tbody>
</table>

F. Casacuberta – DSIC-ITI-UPV

24-28 January 2005

8: 47

Pattern Recognition approaches to Machine Translation

Recursive Alignment Models

Index

1 Introduction ▷ 2

2 A recursive alignment model: MAR ▷ 11

3 Stochastic inversion transduction grammar ▷ 30

4 Bilingual Recursive Alignments ▷ 34

◁ 5 Bibliography ▷ 47
Bibliography

1. H. Ney: *Statistical Natural Language Processing*. STC Doctorate Program, UPC. 2003
Seminars on Formal Syntax and Semantics
Universitat Rovira i Virgili

Pattern Recognition Approaches to Machine Translation
E. Vidal and F. Casacuberta
Pattern Recognition and Human Language Technology Group
Departament de Sistemes Informàtics i Computació
Institut Tecnològic d’Informàtica
Universitat Politècnica de València

9: Speech-to-Speech Translation

Francisco Casacuberta Nolla
fcn@iti.upv.es

24-28 January 2005

F. Casacuberta – DSIC-ITI-UPV

Index

1 Speech processing ▷ 2
2 Automatic speech recognition ▷ 7
3 Speech to speech translation ▷ 19
4 Bibliography ▷ 33
Index

- 1 *Speech processing* \( \Rightarrow \) 2
- 2 Automatic speech recognition \( \Rightarrow \) 7
- 3 Speech to speech translation \( \Rightarrow \) 19
- 4 Bibliography \( \Rightarrow \) 33

An utterance

/por favor, quiero reservar una habitación doble hasta pasado mañana/
Speech technologies

- Speech synthesis: From text to speech
- Speaker recognition/verification: From speech to the speaker identity.
- Dictation: From speech to text.
- Speech summarization: From speech to text.
- Speech categorization: From speech to simple semantic classes.
- Speech understanding: From speech to “semantic” information.
- Dialog processing: From speech to “semantic” information through complex interactions.
- Speech translation: From speech to speech.

Speech recognition, understanding and translation

Speech recognition:
por favor, quiero reservar una habitación doble hasta pasado mañana.

Speech understanding:
ACTION=RESERVATION) (ROOM_TYPE=DOUBLE) (DATE_OF_ENTRANCE=TODAY) (DATE_OF_LEAVING=TODAY+2)

Speech translation:
I want to book a double room until the day after tomorrow, please.
Some speech characteristics

- There are not clear separation between two adjacent words
- The words can be uttered in different ways (also by the same speaker)
- Noise and distortion.
- A speech sentence can not be well formed (grammatically)
Statistical framework for speech recognition

Given an acoustic sequence $v$, search for the sentence $\hat{x}$:

$$\hat{x} = \arg\max_x Pr(x | v)$$

Using the Bayes’ rule

$$\hat{x} = \arg\max_x Pr(x) \cdot Pr(v | x)$$

**STATISTICAL MODELS FOR SPEECH RECOGNITION**

- $Pr(v | x)$: **Acoustic models** (*HIDDEN MARKOV MODELS*)
- $Pr(x)$: **Language model** (*N-GRAMS OR STOCHASTIC GRAMMARS*)

---

**Speech preprocessing**

1. **Utterance**
2. **Window**
3. **FFT**
4. **Mel filters**
5. **Log**
6. **DCT**
7. **Vectors of acoustic features**
Acoustic units

• Words:
  – Include contextual information (coarticulation).
  – Too many units ⇒ difficult training.

• Phoneme:
  – Context dependent (allophones).
  – Few units ⇒ easy training.

• Compromise:
  – Adequate number of units.
  – With some coarticulation information.
  – Proposals: sylabes, semi-sylabes, diphones, contextual phones, ...

Hidden Markov models (HMM)

\[
a_{1,2} \cdot b_2(x_1) \cdot a_{2,2} \cdot b_2(x_2) \cdot a_{2,3} \cdot b_2(x_3) \cdot a_{2,3} \cdot b_3(x_4) \cdot a_{3,3} \cdot b_3(x_5) \\
a_{3,4} \cdot b_4(x_6) \cdot a_{4,4} \cdot b_4(x_7) \cdot a_{4,4} \cdot b_4(x_8) \cdot a_{4,5} \cdot b_5(x_9) \cdot a_{5,5} \cdot b_5(x_{10}) \cdot a_{5,6}
\]
Hidden Markov models (HMM)

- Components of a HMM $\mathcal{M} = \langle Q, E, a, \pi, b \rangle$
  - **Topology**: $Q$: set of states. $E(=\mathbb{R}^d)$: space of acoustic features.
  - **Probabilistic distributions**:
    * between states $(a : Q \times Q \rightarrow [0, 1])$,
    * initial state $(\pi : Q \rightarrow [0, 1])$
    * emission (density) $(b : Q \times E \rightarrow [0, 1])$.
- **Decoding algorithms**: Forward and Backward.
- **An approximation**: Viterbi (+ Beam Search + Histogram Pruning).
- **Training algorithms**:
  - Maximum likelihood Baum-Welch, Viterbi.
  - Other criteria: Maximum mutual information, minimum discriminative information, discriminative.

### Diagram: Training hidden Markov models

- **Sentence**: /la mesa es roja/
- **Utterance**: A sequence of acoustic feature vectors
- **Acoustic-phonetic models**
- **Baum-Welch or Viterbi algorithms**
Word acoustic models

Concatenation of phone units.

/prəˈtɑːr/ 

/a/ /t/ /a/ /r/ 

Language models

Pr(y) = \prod_{i=1}^{I} Pr(y_{i} | y_{i-1}^{i-1})

- Stochastic grammars \( G = (N, \Sigma, R, S, p) \).

Pr(y) \approx P_{G}(y) = \sum_{d(y)} P_{G}(d(y)) \approx \max_{d(y)} P_{G}(d(y))

- N-grams

Pr(y) \approx \prod_{i=1}^{I} p_{n}(y_{i} | y_{i-n+1}^{i-1})

- Learning:
  - Grammatical inference techniques.
  - Maximum likelihood, maximum entropy.
  - Smoothing.
  - Extensions: categories, cache, triggers, etc.
Integrated architecture for speech recognition

\[ \text{Integrated network} \]

\[ \text{Utterance} \xrightarrow{x} \text{Search} \xrightarrow{\arg\max P(s) P(x \mid s)} \text{Decoding} \]

Search engine:
THE VITERBI ALGORITHM (+ beam search + ...)

Integrated architecture for speech decoding

\[ \text{LANGUAGE MODEL} \]

\[ \langle \text{INI} \rangle (1) \quad \langle \text{FIN} \rangle (1) \quad \text{una (0.5)} \quad \text{la (0.5)} \quad \text{bolsa (1)} \quad \text{verde (0.2)} \quad \text{grande (0.8)} \]
An example of speech decoding

\[
\text{argmax} \left( \Pr(x) \cdot \Pr(v|x) \right)
\]
\[\approx \text{“una bolsa verde”}\]

Index

1. Speech processing \(\triangleright\) 2
2. Automatic speech recognition \(\triangleright\) 7
3. Speech to speech translation \(\triangleright\) 19
4. Bibliography \(\triangleright\) 33
Pattern Recognition approaches to Machine Translation

Speech-to-speech translation

General statistical framework for speech translation

Given an acoustic sequence $v$, search for the target sentence $\hat{y}$:

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

The translation can be viewed as:

$$v \rightarrow x \rightarrow y$$

where $x$ is a possible decoding of $v$, and $y$ is the translation of $x$.

$$\arg\max_y \sum_x P_r(y, x \mid v) \approx \arg\max_y \max_x P_r(y, x \mid v)$$


Statistical framework for speech translation

$$\arg\max_y \max_x P_r(y, x \mid v) = \arg\max_y \max_x (P_r(x, y) \cdot P_r(v \mid x))$$

- $P_r(v \mid x)$: Acoustic models
  - HIDDEN MARKOV MODELS
- $P_r(x, y)$: Translation models
  - STOCHASTIC FINITE-STATE TRANSDUCERS

INTEGRATED ARCHITECTURE TO SPEECH TRANSLATION.
Pattern Recognition approaches to Machine Translation

Speech-to-speech translation

Integrated architecture for speech translation

Acoustic models
\[ \Pr(x \mid s) \]

Translation models
\[ P(s,t) \]

Integrated network

Search

\[ \arg\max_{t} \max_{s} P(s,t) \Pr(x \mid s) \]

Search engine:

THE VITERBI ALGORITHM (+ beam search + ...)


Pattern Recognition approaches to Machine Translation

Speech-to-speech translation

Integrated architecture for speech translation

ORIGINAL FINITE-STATE TRANSDUCER

ACOUSTIC MODELS

PHONETIC EXPANSION
An example of speech translation

Argmax \( y, x \) \( Pr(v \mid x) \cdot Pr(y, x) \approx \) “the blue suitcase / la maleta azul”

Statistical framework for speech translation

\[
\text{argmax} \max \Pr(y, x \mid v) = \text{argmax} \max \Pr(y \mid x) \cdot \Pr(x) \cdot Pr(v \mid x)
\]

- \( \Pr(v \mid x) \): Acoustic models
  - \text{HIDDEN MARKOV MODELS}
- \( \Pr(x) \): Source language models
  - \text{N-GRAMS}
- \( \Pr(y \mid x) \): Translation models
  - \text{STOCHASTIC FINITE-STATE TRANSDUCERS}
  - \text{STOCHASTIC ALIGNMENT MODELS + STOCHASTIC DICTIONARIES}

Serial architecture to speech translation.
Serial architecture for speech translation

\[
\text{argmax } \max \{ Pr(y|x) \cdot Pr(x) \cdot Pr(v|x) \}
\]

1. **Word decoding of** \(v\).

\[
\hat{x} = \text{argmax } \{ Pr(x) \cdot Pr(v|x) \}
\]

\(Pr(x)\): source language model; \(Pr(v|x)\): acoustic models.

2. **Translation of** \(\hat{x}\).

\[
\hat{y} = \text{argmax } Pr(y | \hat{x}) = \text{argmax } Pr(y, \hat{x}) = \text{argmax } Pr(\hat{x} | y) \cdot Pr(y)
\]

\(Pr(y, \hat{x})\) or \(Pr(\hat{x} | y)\): translation model; \(Pr(y)\): target language model.
Experimental results with EuTRANS-0 (Spanish to English)

- **Vocabulary**: 686 Spanish words and 513 English words.
- **Text training**: 490,000 pairs (4,655,000/4,802,000 running words)
- **Speech training**: 11,000 running words for 25 CDHMM of monophones.
- **Speech test**: 336 sentences (3,000 running words) (PP=6.8)
- **Source language models for the serial architecture**: trigrams.

<table>
<thead>
<tr>
<th>Models</th>
<th>Architecture</th>
<th>Source Language Model</th>
<th>WER(%)</th>
<th>TWER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMEGA</td>
<td>Integrated</td>
<td>OMEGA</td>
<td>8.4</td>
<td>7.6</td>
</tr>
<tr>
<td>OMEGA</td>
<td>Serial</td>
<td>Trigrams</td>
<td>8.6</td>
<td>9.4</td>
</tr>
<tr>
<td>GIATI</td>
<td>Integrated</td>
<td>GIATI</td>
<td>7.5</td>
<td>10.7</td>
</tr>
<tr>
<td>GIATI</td>
<td>Serial</td>
<td>Trigrams</td>
<td>8.6</td>
<td>11.6</td>
</tr>
<tr>
<td>ALTEMP</td>
<td>Serial</td>
<td>Trigrams</td>
<td>8.6</td>
<td>9.9</td>
</tr>
</tbody>
</table>

Experimental results with EuTRANS-II (Italian to English)

- **Vocabulary**: 2,459 Italian words and 1,701 English words.
- **Text training**: 3,038 pairs (61,232/72,446 running words)
- **Speech training**: 52,511 running words for 2,700 CDHMM of triphones.
- **Speech test**: 278 sentences (5,381 running words) (PP=6.8)
- **Source language models for the serial architecture**: trigrams.

<table>
<thead>
<tr>
<th>Models</th>
<th>Architecture</th>
<th>Source Language Model</th>
<th>WER(%)</th>
<th>TWER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIATI</td>
<td>Serial</td>
<td>Trigrams</td>
<td>22.1</td>
<td>37.9</td>
</tr>
<tr>
<td>GIATI</td>
<td>Integrated</td>
<td>GIATI</td>
<td>32.0</td>
<td>44.8</td>
</tr>
<tr>
<td>OMEGA</td>
<td>Serial</td>
<td>Trigrams</td>
<td>22.1</td>
<td>49.4</td>
</tr>
<tr>
<td>OMEGA</td>
<td>Integrated</td>
<td>OMEGA</td>
<td>52.5</td>
<td>57.0</td>
</tr>
<tr>
<td>ALTEMP</td>
<td>Serial</td>
<td>Trigrams</td>
<td>22.1</td>
<td>37.8</td>
</tr>
</tbody>
</table>
Iterative search (1)

\[
\arg\max_y \Pr(y | v) \approx \arg\max_y \max_x \Pr(y) \cdot \Pr(x | y) \cdot \Pr(v | x)
\]

a) INITIALIZATION

1. **Decoding v:**
   \[\hat{x} \approx \arg\max_x \{ \Pr(x) \cdot \Pr(v | x) \} \]

2. **Translating \( \hat{x} \):**
   \[\hat{y} \approx \arg\max_y \Pr(\hat{x} | y) \cdot \Pr(y)\]

b) GENERAL ITERATION

1. **Decoding v using \( \hat{y} \):**
   \[\hat{x} \approx \arg\max_x \{ \Pr(x | \hat{y}) \cdot \Pr(v | x) \} \]

2. **Translating \( \hat{x} \):**
   \[\hat{y} \approx \arg\max_y \Pr(\hat{x} | y) \cdot \Pr(y)\]
On-line demos
http://prhltdemos.iti.es/demo/

Index

1 Speech processing ▶ 2
2 Automatic speech recognition ▶ 7
3 Speech to speech translation ▶ 19
4 Bibliography ▶ 33
Bibliography

Pattern Recognition approaches to Machine Translation

F. Casacuberta and E. Vidal
Pattern Recognition and Human Language Technology Group
Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Spain

Computer Assisted Translation

Enrique Vidal
evidal@iti.upv.es

January 2005

E. Vidal – ITI-UPV-DSIC

Index

1 Computer Assisted Translation (CAT) ▷ 2
2 Statistical Framework for (text-input) CAT ▷ 7
3 Interactive Search ▷ 9
4 Using Speech in the CAT Framework ▷ 22
5 Bibliography ▷ 31
Introduction to Computer Assisted Translation (CAT)

- MT systems are not perfect: they often produce erroneous (portions) of target-language text
- To correct these errors, human post-processing is generally needed
- CAT aims to increase the overall (MT + human) productivity by incorporating human correction activities within the translation process itself

**Main idea:**

*Iterative process where human activity is embedded in the loop*

- Use a MT system to produce target text segments that can be accepted or amended by a human translator; these correct(ed) segments are then used by the MT system as additional information to achieve further, hopefully improved suggestions
**CAT Human-Machine (keyboard) interactive process**

- In each iteration, a correct prefix ($y_p$) of the target sentence is available and the CAT system computes its best (or $N$-best) translation suffix hypothesis ($\hat{y}_s$) to complete this prefix.
- Given $y_p\hat{y}_s$, the CAT cycle proceeds by letting the user establish a new, longer acceptable prefix.

This prefix is typically formed by $y_p$, followed by an initial part of $\hat{y}_s$ accepted by the user ($a$), followed by text obtained by means of additional user keystrokes ($k$) generally aimed to amend remaining incorrect parts of $\hat{y}_s$.

This prefix becomes a new $y_p$, thereby starting a new CAT prediction cycle

- Ergonomics and user preferences dictate exactly when the system can start its new cycle, but typically, it is started after each user-entered word or even after each new user keystroke.
- These ideas were studied in [Foster02] and have been thoroughly explored in the TT2 project

---

**CAT human-machine (keyboard) interactive process: example**

Translating the source sentence “Click OK to close the print dialog” into Spanish:

<table>
<thead>
<tr>
<th>ITER-0</th>
<th>($y_p$)</th>
<th>()</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER-1</td>
<td>($\hat{y}_s$)</td>
<td>(Haga clic para cerrar el diálogo de impresión)</td>
</tr>
<tr>
<td></td>
<td>($a$)</td>
<td>(Haga clic)</td>
</tr>
<tr>
<td></td>
<td>($k$)</td>
<td>(en)</td>
</tr>
<tr>
<td></td>
<td>($y_p$)</td>
<td>(Haga clic en)</td>
</tr>
<tr>
<td>ITER-2</td>
<td>($\hat{y}_s$)</td>
<td>(ACEPTAR para cerrar el diálogo de impresión)</td>
</tr>
<tr>
<td></td>
<td>($a$)</td>
<td>(ACEPTAR para cerrar el)</td>
</tr>
<tr>
<td></td>
<td>($k$)</td>
<td>(cuadro)</td>
</tr>
<tr>
<td></td>
<td>($y_p$)</td>
<td>(Haga clic en ACEPTAR para cerrar el cuadro)</td>
</tr>
<tr>
<td>FINAL</td>
<td>($\hat{y}_s$)</td>
<td>(de diálogo de impresión)</td>
</tr>
<tr>
<td></td>
<td>($a$)</td>
<td>(de diálogo de impresión)</td>
</tr>
<tr>
<td></td>
<td>($k$)</td>
<td>(interaction)</td>
</tr>
<tr>
<td></td>
<td>($y_p \equiv y$)</td>
<td>(Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión)</td>
</tr>
</tbody>
</table>

System suggestions are printed in cursive and user input in boldface typewriter font.

In the final translation, $y$, text that have been typed by the user is underlined
Evaluating MT and CAT systems

THREE MEASURES

- **TRANSLATION WORD ERROR RATE (TWER):**
  Minimum number of word insertions, deletions and substitutions needed to edit the system output into a (single) target reference

- **TRANSLATION CHARACTER ERROR RATE (TWER):**
  Minimum number of character insertions, deletions and substitutions needed to edit the system output into a (single) target reference

- **KEY-STROKE RATIO (KSR):**
  Number of key-strokes that are necessary to achieve a (single) target reference divided by the number of running characters.
Text prediction for Computer-Assisted Translation (CAT)

Given a source text $x$ and a "correct" prefix $y_p$ of the target text, search for a suffix $\hat{y}_s$, that maximizes the posterior probability over all possible suffices:

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

Taking into account that $Pr(y_p | x)$ does not depend on $y_s$, we can write:

$$\hat{y}_s = \arg\max_{y_s} Pr(y_p y_s | x)$$
$$= \arg\max_{y_s} Pr(x | y_p y_s) \cdot Pr(y_p y_s) \quad (1)$$
$$= \arg\max_{y_s} Pr(x, y_p y_s) \quad (2)$$

- (1): Statistical Alignment and Language models
- (2) Stochastic Finite State Transducers
- Text-input MT is a particular case, where $y_p = \lambda$
- Main difference of CAT vs. MT: search over the set of suffixes

Index

1 Computer Assisted Translation (CAT) ▷ 2
2 Statistical Framework for (text-input) CAT ▷ 7
3 Interactive Search ▷ 9
4 Using Speech in the CAT Framework ▷ 22
5 Bibliography ▷ 31
CAT Interactive Search

High speed is needed because typically a new system hypothesis must be produced in real time after each user keystroke.

**WORD-GRAF BASED APPROACH:**

- For each source sentence, a graph representing all its possible translations according to the translation model is generated.
- In each CAT iteration, the Word-Graph is searched for a best path compatible with the prefix given in this iteration.
- Error-Correcting smoothing (edit distance) is used to allow for user-given prefixes that may not exist in the Word-Graph.
- Computation is carried out in an incremental manner: in each iteration the results from the previous iteration are updated.

Example of CAT human-machine (keyboard) interaction

**S:** Load your originals into the Document Feeder

**H:** Cargue los originales en la
Example of CAT human-machine (keyboard) interaction

S: Load your originals into the Document Feeder

H: Cargue los originales en la alimentador de originales

P: Cargue los originales en e
Example of CAT human-machine (keyboard) interaction

**S:** Load your originals into the Document Feeder

**H:** Cargue los originales en la

**P:** Cargue los originales en e

**H:** Cargue los originales en el alimentador de originales

**T:** Cargue los originales en el alimentador de originales

**S:** Source sentence \((x)\)
**P:** Current human-validated Prefix \((y_p)\)
**H:** System Hypothesis \((\hat{y}_s)\)
**T:** Final Translation

More examples of CAT human-machine (keyboard) interaction

**S:** It also contains a section to help users of previous software versions adapt more quickly to the new software

**H:** Se se para ayudar a los usuarios de versiones anteriores del software a que se a dapten más rápidamente a este nuevo software

**P:** T

**H:** También se ofrece una sección para ayudar a los usuarios de versiones anteriores del software a que se adapt en más rápidamente a este nuevo software

**P:** También c

**H:** También contiene una sección para ayudar a los usuarios de versiones anteriores del software a que se adapten más rápidamente a este nuevo software

**T:** También contiene una sección para ayudar a los usuarios de versiones anteriores del software a que se adapten más rápidamente a este nuevo software
More examples of CAT human-machine (keyboard) interaction

S: Dirección de la alimentación para tamaños de papel estándar 1-9

H: Feed direction for standard stock names 1-9

P: Feed direction for standard paper

H: Feed direction for standard paper sizes 1-9

T: Feed direction for standard paper sizes 1-9

S: Edición de la lista de impresoras

H: Editing printers

P: Editing the printers

H: Editing the printer

T: Editing the printer list

S: Edición de la lista de impresoras

H: Editing the printer list

T: Editing the printer list
## Benchmark Xerox printer manuals corpus

<table>
<thead>
<tr>
<th>Data</th>
<th>English</th>
<th>Spanish</th>
<th>English</th>
<th>German</th>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>Sent. pairs</td>
<td>56K</td>
<td>53K</td>
<td>49K</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run. words</td>
<td>572K</td>
<td>657K</td>
<td>543K</td>
<td>583K</td>
<td>507K</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>26K</td>
<td>30K</td>
<td>25K</td>
<td>27K</td>
<td>25K</td>
</tr>
<tr>
<td>Test</td>
<td>Sentences</td>
<td>1 125</td>
<td>984</td>
<td>996</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run. words</td>
<td>7.6K</td>
<td>9.4K</td>
<td>9.6K</td>
<td>10.0K</td>
<td>10.8K</td>
</tr>
<tr>
<td></td>
<td>Out of Voc.</td>
<td>341</td>
<td>362</td>
<td>219</td>
<td>552</td>
<td>252</td>
</tr>
<tr>
<td></td>
<td>Run. chars.</td>
<td>46K</td>
<td>58K</td>
<td>55K</td>
<td>63K</td>
<td>61K</td>
</tr>
<tr>
<td></td>
<td>Perplexity</td>
<td>107</td>
<td>60</td>
<td>93</td>
<td>169</td>
<td>193</td>
</tr>
</tbody>
</table>

## Benchmark EU bulletin corpus

<table>
<thead>
<tr>
<th>Data</th>
<th>English</th>
<th>Spanish</th>
<th>English</th>
<th>German</th>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>Sent. pairs</td>
<td>214K</td>
<td>223K</td>
<td>215K</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run. words</td>
<td>5.9M</td>
<td>6.6M</td>
<td>6.5M</td>
<td>6.1M</td>
<td>6.0M</td>
</tr>
<tr>
<td></td>
<td>Vocabulary</td>
<td>84K</td>
<td>97K</td>
<td>87K</td>
<td>152K</td>
<td>85K</td>
</tr>
<tr>
<td>Test</td>
<td>Sentences</td>
<td>800</td>
<td>800</td>
<td>800</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run. words</td>
<td>20K</td>
<td>25K</td>
<td>22K</td>
<td>21K</td>
<td>22K</td>
</tr>
<tr>
<td></td>
<td>Out of Voc.</td>
<td>108</td>
<td>140</td>
<td>107</td>
<td>227</td>
<td>113</td>
</tr>
<tr>
<td></td>
<td>Perplexity</td>
<td>96</td>
<td>72</td>
<td>95</td>
<td>153</td>
<td>97</td>
</tr>
</tbody>
</table>
## CAT results with the Xerox corpus

<table>
<thead>
<tr>
<th>DATA:</th>
<th>GIATI 3-gram (1-best)</th>
<th>GIATI 3-gram (5-best)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KSR</td>
<td>CER</td>
</tr>
<tr>
<td>XRCE2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>En-Es</td>
<td>17.6</td>
<td>30.3</td>
</tr>
<tr>
<td>Es-En</td>
<td>21.5</td>
<td>35.5</td>
</tr>
<tr>
<td>En-Fr</td>
<td>37.1</td>
<td>54.3</td>
</tr>
<tr>
<td>Fr-En</td>
<td>39.4</td>
<td>55.3</td>
</tr>
<tr>
<td>En-De</td>
<td>38.8</td>
<td>62.8</td>
</tr>
<tr>
<td>De-En</td>
<td>36.4</td>
<td>61.5</td>
</tr>
</tbody>
</table>

## CAT results with the EU corpus

<table>
<thead>
<tr>
<th>DATA:</th>
<th>GIATI 5-gram (1-best)</th>
<th>GIATI 5-gram (5-best)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KSR</td>
<td>CER</td>
</tr>
<tr>
<td>EU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>En-Es</td>
<td>27.5</td>
<td>37.6</td>
</tr>
<tr>
<td>Es-En</td>
<td>25.4</td>
<td>38.0</td>
</tr>
<tr>
<td>En-Fr</td>
<td>26.2</td>
<td>36.0</td>
</tr>
<tr>
<td>Fr-En</td>
<td>23.1</td>
<td>36.1</td>
</tr>
<tr>
<td>En-De</td>
<td>29.4</td>
<td>41.2</td>
</tr>
<tr>
<td>De-En</td>
<td>31.0</td>
<td>44.4</td>
</tr>
</tbody>
</table>
Using Speech Recognition in CAT

- Early idea: a human translator dictates aloud the translation in the target language. As the source text is known by the system, this knowledge can be used to reduce recognition errors.

- Alternative idea within the CAT framework: the human translator determines acceptable prefixes of the suggestions made by the system by reading (with possible modifications) parts of these suggestions.
  - A much lower degree of freedom is possible and the correspondingly lower perplexity allows for sufficiently high recognition accuracy.
  - As this is fully integrated within the CAT paradigm, the user can make use of the conventional means (keyboard and/or mouse) to guarantee that the produced text exhibits an adequate level of quality.
Title: Target language dictation in CAT

A human translator dictates the translation of a source text, \( x \), producing a target language acoustic sequence \( v \).

Given \( v \) and \( x \), the system should search for a most likely decoding of \( v \):

\[
\hat{y} = \underset{y}{\text{argmax}} \Pr(y \mid x, v)
\]

By the assumption that \( \Pr(v \mid x, y) \) does not depend on \( x \),

\[
\hat{y} = \underset{y}{\text{argmax}} \Pr(v \mid y) \cdot \Pr(x \mid y) \cdot \Pr(y)
\]

- \( \Pr(v \mid y) \approx (\text{TARGET LANGUAGE ACOUSTIC MODELS}) \)
- \( \Pr(x \mid y) \approx \text{TRANSLATION MODEL} \)
- \( \Pr(y) \approx \text{TARGET LANGUAGE MODEL} \)

**Similar to plain speech decoding, where:**

\[
\hat{y} = \underset{y}{\text{argmax}} \Pr(v \mid y) \cdot \Pr(y)
\]

### Further use of speech recognition in CAT

Let \( x \) be the source text and \( y_p \) a “correct” prefix of the target sentence. As in pure text CAT the system suggests an optimal suffix:

\[
\hat{y}_s = \underset{y_s}{\text{argmax}} \Pr(y_s \mid x, y_p)
\] (3)

The user is now allowed to utter some words, \( v \), generally aimed at amending parts of \( \hat{y}_s \) and the system has then to obtain a most probable decoding of \( v \):

\[
\hat{d} = \underset{d}{\text{argmax}} \Pr(d \mid x, y_p, \hat{y}_s, v)
\] (4)

Finally, the user can enter additional amendment keystrokes \( k \), to produce a new consolidated prefix, \( y_p \), based on the previous \( y_p, \hat{d}, k \) and parts of \( \hat{y}_s \).
Example of speech-enabled CAT human-machine interaction

Translating the source sentence “Click OK to close the print dialog” into Spanish:

<table>
<thead>
<tr>
<th>ITER-0</th>
<th>((y_p))</th>
<th>()</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER-1</td>
<td>((y_s))</td>
<td>(Haga clic para cerrar el diálogo de impresión)</td>
</tr>
<tr>
<td></td>
<td>((v))</td>
<td>(Haga clic a)</td>
</tr>
<tr>
<td></td>
<td>((d))</td>
<td>(en ACEPTAR)</td>
</tr>
<tr>
<td></td>
<td>((k))</td>
<td>(Haga clic en ACEPTAR)</td>
</tr>
<tr>
<td>ITER-2</td>
<td>((y_s))</td>
<td>(para cerrar el diálogo de impresión)</td>
</tr>
<tr>
<td></td>
<td>((v))</td>
<td>(cerrar el cuadro)</td>
</tr>
<tr>
<td></td>
<td>((k))</td>
<td>()</td>
</tr>
<tr>
<td></td>
<td>((y_p))</td>
<td>(Haga clic en ACEPTAR para cerrar el cuadro)</td>
</tr>
<tr>
<td>FINAL</td>
<td>((y_s))</td>
<td>(de diálogo de impresión)</td>
</tr>
<tr>
<td></td>
<td>((k))</td>
<td>(#)</td>
</tr>
<tr>
<td></td>
<td>((y_p = y))</td>
<td>(Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión)</td>
</tr>
</tbody>
</table>

System suggestions are printed in cursive, text decoded from user speech in boldface and typed text in boldface typewriter font. In the final translation, \(y\), text obtained from speech decoding is marked in boldface, while typed text is underlined.

Models for speech recognition in CAT

From Eq. (4):
\[
\hat{d} = \arg\max_{d} \Pr(d \mid x, y_p, \hat{y}_s, v) = \arg\max_{d} \Pr(d \mid x, y_p, \hat{y}_s) \cdot \Pr(v \mid x, y_p, \hat{y}_s, d)
\]

and, by making the assumption that \(\Pr(v \mid x, y_p, \hat{y}_s, d)\) only depends on \(d\):

\[
\hat{d} = \arg\max_{d} \Pr(d \mid x, y_p, \hat{y}_s) \cdot \Pr(v \mid d)
\]

- \(\Pr(v \mid d) \approx \text{(TARGET LANGUAGE) ACOUSTIC MODELS}\)
- \(\Pr(d \mid x, y_p, \hat{y}_s) \approx \text{TARGET LANGUAGE MODEL CONSTRAINED BY THE SOURCE SENTENCE, THE PREFIX AND THE SUFFIX}\)

Less and more restricted scenarios, depending on the latter model:
- CAT–PREF: Ignore the dependency on the system suggestion \(\hat{y}_s\)
- CAT–SEL: Restrict \(d\) to be just a prefix of \(\hat{y}_s\)
Speech recognition in CAT: CAT–PREF

Starting from:

\[ \hat{d} = \arg \max_d \Pr(d \mid x, y_p, \hat{y}_s) \cdot \Pr(v \mid d) \]

A less restricted scenario arises if only the prefix \( y_p \) is available; that is, the previous system prediction \( \hat{y}_s \) is ignored and the user is assumed to produce free target speech, only constrained to be a translation of the source text and a continuation of the given prefix:

\[ \hat{d} = \arg \max_d \Pr(d \mid x, y_p) \cdot \Pr(v \mid d) \]

As compared with the dictated-translation framework, this adds the constraint provided by the target text prefix, \( y_p \), thereby allowing for higher speech decoding accuracy.

Most restricted speech recognition in CAT: CAT–SEL

Starting from:

\[ \hat{d} = \arg \max_d \Pr(d \mid x, y_p, \hat{y}_s) \cdot \Pr(v \mid d) \]

A most restricted scenario appears if the decoding of \( v \) is constrained to be exactly a prefix of the suffix suggested by the system, \( \hat{y}_s \).

The uttered prefix would help the user determine an accepted part of the system suggestion.

In this case, \( \Pr(d \mid x, y_p, \hat{y}_s) = \Pr(d \mid \hat{y}_s) \) and the above equation can be written as:

\[ \hat{d} = \arg \max_d \Pr(d \mid \hat{y}_s) \cdot \Pr(v \mid d) \]

As compared with all the previous scenarios involving speech, here \( \Pr(d \mid \hat{y}_s) \) can be modeled by a very low perplexity language model, which allows for much higher speech decoding accuracy.
CAT speech recognition results

- SPEECH DATA: Utterances of fragments of target language sentences from the test XEROX CORPUS (485 fragments, 10 speakers, 5,796 utterances)
- MODELS: derived from both source and target sentences of the training XEROX corpus
- DEC and DEC-PREF used for comparison:
  - DEC: Conventional speech recognition of target language utterances (source text ignored)
  - DEC-PREF: Target speech recognition constrained by the known prefix (source text ignored)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>DEC-PREF</th>
<th>CAT-PREF</th>
<th>CAT-SEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Error Rate (%)</td>
<td>18.6</td>
<td>16.1</td>
<td>10.6</td>
<td>1.6</td>
</tr>
<tr>
<td>Sentence Error rate (%)</td>
<td>50.2</td>
<td>44.4</td>
<td>30.0</td>
<td>3.6</td>
</tr>
</tbody>
</table>

Using knowledge about the source sentence is more important than using only user-validated prefixes
Bibliography


