Computational Intelligence and Learning Doctoral School
U. C. Louvain

Machine Translation:
Finite-State Models and Statistical Approaches

Enrique Vidal

Pattern Recognition and Human Language Technology Group

Instituto Tecnológico de Informática

Departamento de Sistemas Informáticos y Computación

Universisdad Politécnica de Valencia, Spain

September 2007

E.Vidal – ITI-UPV-DSIC

Index

Computational Intelligence and Learning Doctoral School
U. C. Louvain

Machine Translation:
Finite-State Models and Statistical Approaches

Introduction

Enrique Vidal

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

September 2007

E. Vidal – ITI-UPV-DSIC

Index

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

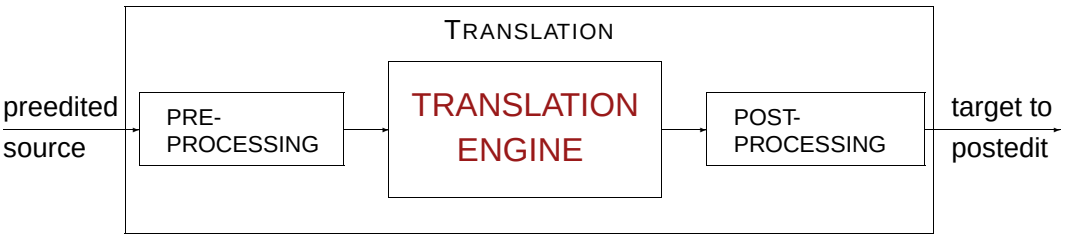
- MT is useful.
- There are many situations that MT systems produce reliable, if less than perfect, translations at high speed.
- In some circunstances, MT systems can produce good quality outputs.
- MT does not threaten transaltors' jobs: High demand of translations and too repetitive translation jobs.
- 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

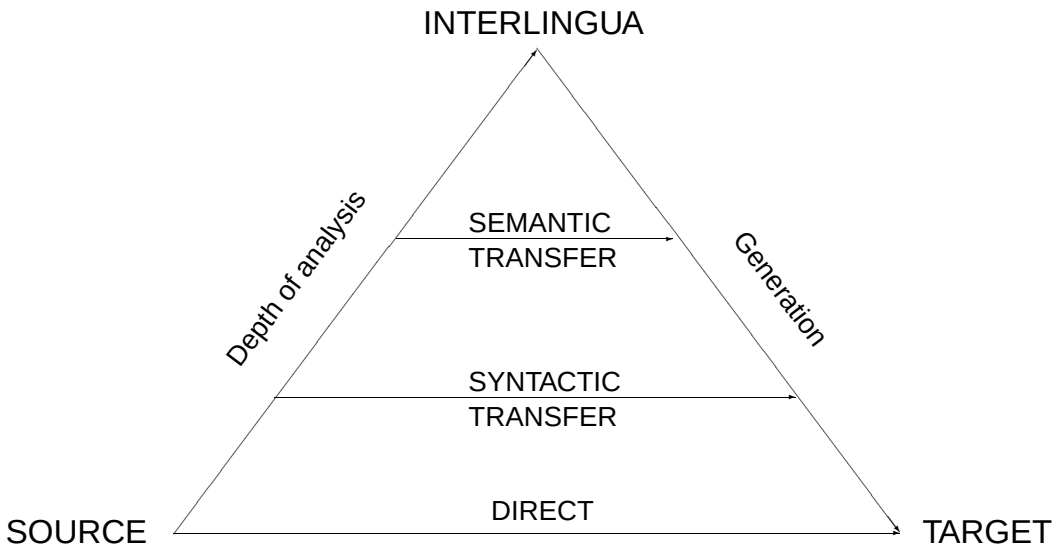
- While the number of errors and bad constructions is high, “post-editing” can make the result useful.
- Many problems could have been avoided by making the source text “simpler”.

Need of pre/post-processing

- Tokenization
- Case normalization
- Named entities or “Lexical categorization”
- Phrase detection
- Sublanguages (dates, time of day, etc.)



Approaches to MT: Analysis detail



Approaches to MT: Technologies

- (Linguistic) knowledge-based methods
- (Memorized) example-based methods
  - Translation memories
- Statistical models
  - Alignment models
  - Finite-state models
  - Neural networks
- Hybrid models

Linguistic resources and tools

- Dictionaries
- Morpho-syntactic taggers
- Shallow parsers
- Chunkers
- Grammars
- Corpora
- Paragraph-aligned and labeled corpora

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, ...

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.

Assessment

- Only test sentences
  - Subjective evaluation based on the number of words that need to be corrected or deleted

- Test sentences with reference translation
  - Automatic assessment
    - Edit Distances: Translation Word Error Rate (WER)
    - Multireference WER (mWER)
    - Position-independent WER (PER)
    - N-gram based: *BLUE* and *NIST* score
  - Assessment for CAT
    - Key Stroke Ratio (KSR)
    - “Mouse Action” ratio (MAR)
    - “Word Stroke” Ratio (WSR)

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
- **1995-2006** Development of statistical techniques and other empiricists methods
  - Progress of the statistical approach: from word-based to phrase-based models; log-linear combination of models, etc.
  - 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

Computational Intelligence and Learning Doctoral School
U. C. Louvain

Machine Translation:
Finite-State Models and Statistical Approaches

Software Systems and Tools

Enrique Vidal

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

September 2007

E.Vidal – ITI-UPV-DSIC

Index

Web text translation systems

- SYSTRAN: `www.systransoft.com/index.html`
- ALTAVISTA (SYSTRAN): `http://babelfish.altavista.com`
- Google: `www.google.es/language_tools?hl=es`
- IBM WebSphere: `http://www-306.ibm.com/software/pervasive/tech/demos/translation.shtml`
- WorldLingo: `www.worldlingo.com/en/products_services/worldlingo_translator.html`
- Free2ProfessionalTranslation: `www.freetranslation.com`
- TranslateNow: `www.foreignword.com/Tools/transnow.htm`
- Prompt-Online: `http://translation2.paralink.com`
- Phrase-based systems: `http://dcomgp05.gnd.upv.es/WebTrans.debug/root`
- interNOSTRUM: `www.internostrum.com`
- SisHiTra: `http://prhltdemos.iti.upv.es/~sishitra`

Office text translation systems

- SYSTRAN: `www.systransoft.com/index.html`
- SALT: `www.cult.gva.es/salt/salt_programes_salt2.htm`
- WebSphere: `http://www-306.ibm.com/software/pervasive/tech/demos/translation.shtml`

Computer-assisted translation systems

- TRANSTYPE:
  `http://rali.iro.umontreal.ca/Transtype2/Demo/index.en.html`
- Déjàvu:
  `www.atril.com`
- TRADOS:
  `www.trados.com/products.asp?page=1213`
- Sakhr Translator Workbench:
  `http://aramedia.com/catrans.htm`

Speech translation systems

- EUTRANS:
  `http://prhltdemos.iti.es/demo`
- Anuvaad (ATT):
  `www.research.att.com/~srini/Projects/Anuvaad/home.html`
- IBM ViaVoice Translator 1.0:
  `http://domino.research.ibm.com/comm/research.nsf/pages/r.uit.innovation.html`

Index

Bilingual dictionaries and statistical language modelling

- List of dictionaries
  `www.yourdictionary.com/languages.html`
- WordReference
  `www.wordreference.com/es/Index.htm`
- LexiCool
  `www.lexicool.com/`
- TravLang
  `http://dictionaries.travlang.com`
- SRLIM: The SRI Language Modeling Toolkit
  `www.speech.sri.com/projects/srilm`

Statistical translation tools

- GIZA: Training statistical translation models
  `http://www.fjoch.com/GIZA++.html`
- The ISI ReWrite
  `http://www.isi.edu/licensed-sw/rewrite-decoder/`
- Pharaoh
  `http://www.isi.edu/publications/licensed-sw/pharaoh/`
- Thot: Learning phrase-based MT
  `http://thot.sourceforge.net/`
- Moses Phrase-based MT tool-kit
  `http://www.statmt.org/moses/`

Index

Giza tool-kits

- The EGYPT Statistical Machine Translation Toolkit contains GIZA, a training program that learns statistical translation models from bilingual corpora. GIZA is written C++ with the STL library (tested using gnu C++).

  `www.clsp.jhu.edu/ws99/projects/mt/toolkit/`

  (Developped in WS'99 Summer Workshop organized by the Center for Language and Speech Processing of the the Johns Hopkins University)

- GIZA++ is an extension of GIZA

  Original version: `www.fjoch.com/GIZA++.html`

  Patched version: `http://ling.umd.edu/~redpony/software/`

  GIZA++ is used today to obtain word alignments in a bilingual corpus. These alignments are the basis to build *phrase-based models*, the state of the art in SMT.

GIZA++ Package Programs

- GIZA++: GIZA++ itself
- plain2snt.out: simple tool to transform plain text into GIZA format
- plain2snt.out: simple tool to transform GIZA format into plain text
- trainGIZA++.sh:
  Shell script to perform standard training given a corpus in GIZA format
- mkcls: Computes word classes in a monolingual corpus
- snt2cooc: Generates a coocurrence file

Input File Formats: vocabulary files

Each entry is stored on one line as follows:

```
uniq_id1 string1 no_occurrences1
uniq_id2 string2 no_occurrences2
uniq_id3 string3 no_occurrences3
...
```

Example:

Source vocabulary file	Target vocabulary file
...	...
176 desierto 8	731 eleccions 33
177 fueron 61	732 article 16
178 comprobar 6	733 nostra 23
179 instalaciones 15	734 alternativa 12
180 superado 4	735 contundent 3
...	...

uniq_ids are sequential positive integer numbers.
0 is reserved for the special token NULL.

Input File Formats: bitext files

Each sentence pair is stored in three lines:

- The first line is the number of times of the sentence pair.
- The second line is the source sentence coded using the vocabulary file and
- the third is the target sentence in the same format.

A sample of 3 pairs:

...
1
119 109 120 20 121 122 7 123 124 29 72 125 126 57 22 127 128 129 10 11 12
63 29 3 129 9 130 131 8 132 133 55 78 134 135 60 124 136 137 66 9 13 12 14
1
130 131 132
138 139 140
1
114 133 134 12
123 8 141 142 14
...

Input File Formats: dictionary File

- The dictionary file is is optional. Format:

  `target_word_id source_word_id`

- The list should be sorted by the target_word_id.

- If a dictionary is provided in in the configuration file, GIZA++ will change the cooccurrence counting in the first iteration of model 1 to honor the so-called "Dictionary Constraint":

Output file formats: probability tables

1. Translation table (`*.t*.*`)

   ```
   prob_table.t1.n = t table after n iterations of Model1 training
   prob_table.t2.n = t table after n iterations of Model2 training
   prob_table.t2to3 = t table after transfering Model2 to Model3
   prob_table.t3.n = t table after n iterations of Model3 training
   prob_table.t4.n = t table after n iterations of Model4 training
   ```

   Each line is of the following format:

   $$s\_id \;\; t\_id \;\; P(t\_id|s\_id)$$

2. Fertility table (`*.n3.*`)

   Each line in this file is of the following format:

   $$source\_token\_id \;\; p0 \;\; p1 \;\; p2 \;\; \ldots \;\; pn$$

   where $p0$ is the probability that the source token has zero fertility; $p1$, fertility one, ..., and n is the maximum possible fertility as defined in the program.

Output file formats: probability tables

3. Probability of inserting a null after a source word (`*.p0*`)

   Contains only one line with the probability of not inserting a NULL token.

4. Alignment tables (`*.a*.*`)

   The format of each line is as follows:

   $$i \;\; j \;\; l \;\; m \;\; P(i \mid j,\, l,\, m)$$

   where:

   j = position in target sentence i = position in source sentence
   l = length of source sentence m = length of target sentence

   and $P(i \mid j, l, m)$ is the probability that a source word in position i is moved to position j in a pair of sentences of length l and m.

5. Distortion table (`*.d3.*`)

   The format is similar to the alignment tables but the positions of i and j are swapped:

   $$j \;\; i \;\; l \;\; m \;\; P(j \mid i,\, l,\, m)$$

Output file formats: probability tables

6. Distortion table for IBM-4 (`*.d4.*`)
7. Distortion table for IBM-5 (`*.d5.*`)
8. Alignment probability table for HMM alignment mode (`*.A3.*`)
9. Perplexity File (`*.perp`)
10. Revised vocabulary files (`*.src.vcb, *.trg.vcb`)
11. Final parameter file: (`*.gizacfg`)

Config file for GIZA++

```
// general parameters:
// -------------------
ml  101    (maximum sentence length)

// No. of iterations:
// -------------------
hmmiterations  5      (mh)
model1iterations  5   (number of iterations for Model 1)
model2iterations  0   (number of iterations for Model 2)
model3iterations  5   (number of iterations for Model 3)
model4iterations  5   (number of iterations for Model 4)
model5iterations  0   (number of iterations for Model 5)
model6iterations  0   (number of iterations for Model 6)

// parameter for various heuristics in GIZA++ for efficient training:
// ------------------------------------------------------------------
countincreasecutoff  1e-06   (Counts increment cutoff threshold)
countincreasecutoffal   1e-05
// (Counts increment cutoff threshold for alignments in training of
// fertility models)
mincountincrease   1e-07   (minimal count increase)
peggedcutoff   0.03
```



```
// (relative cutoff probability for alignment-centers in pegging)
probcutoff   1e-07   (Probability cutoff threshold for lexicon probabilities)
probsmooth   1e-07   (probability smoothing (floor) value )

// parameters for describing the type and amount of output:
// --------------------------------------------------------
compactalignmentformat   0
// (0: detailled alignment format, 1: compact alignment format )
hmmdumpfrequency   0   (dump frequency of HMM)
// l      (log file name)
log   0  (0: no logfile; 1: logfile)
model1dumpfrequency   0  (dump frequency of Model 1)
model2dumpfrequency   0  (dump frequency of Model 2)
model345dumpfrequency   0  (dump frequency of Model 3/4/5)
nbestalignments   0  (for printing the n best alignments)
nodumps   0  (1: do not write any files)
// o     (output file prefix)
onlyaldumps   0  (1: do not write any files)
// outputpath     (output path)
transferdumpfrequency   0  (output: dump of transfer from Model 2 to 3)
verbose   0  (0: not verbose; 1: verbose)
verbosesentence   -10
// (number of sentence for which a lot of information should be printed
// (negative: no output))
```

```
// parameters describing input files:
// ---------------------------------
// c     (training corpus file name)
// d     (dictionary file name)
// s     (source vocabulary file name)
// t     (target vocabulary file name)
// tc     (test corpus file name)

// smoothing parameters:
// --------------------
emalsmooth   0.2
// (f-b-trn: smoothing factor for HMM alignment model (can be
// ignored by -emSmoothHMM))
model23smoothfactor   0
// (smoothing parameter for IBM-2/3 (interpolation with constant))
model4smoothfactor   0.2
// (smooting parameter for alignment probabilities in Model 4)
model5smoothfactor   0.1
// (smooting parameter for distortion probabilities in Model 5
// (linear interpolation with constant))
nsmooth   64
// (smoothing for fertility parameters (good value: 64):
// weight for wordlength-dependent fertility parameters)
```



```
nsmoothgeneral   0
//(smoothing for fertility parameters (default: 0): weight for
// word-independent fertility parameters)

// parameters modifying the models:
// --------------------------------
compactadtable   1
// (1: only 3-dimensional alignment table for IBM-2 and IBM-3)
deficientdistortionforemptyword   0
// (0: IBM-3/IBM-4 as described in (Brown et al. 1993);
//  1: distortion model of empty word is deficient;
//  2: distoriton model of empty word is deficient (differently);
//     setting this parameter also helps to avoid that during IBM-3
//     and IBM-4 training too many words are aligned with the empty word)
depm4   76
// (d_{ 1}: &1:l, &2:m, &4:F, &8:E, d_{>1}&16:l, &32:m, &64:F, &128:E)
depm5   68
// (d_{ 1}: &1:l, &2:m, &4:F, &8:E, d_{>1}&16:l, &32:m, &64:F, &128:E)
emalignmentdependencies   2
// (lextrain: dependencies in the HMM alignment model.
//     &1:  sentence length;
//     &2: previous class;
//     &4: previous position;
//     &8: French position;
```

```
//     &16: French class)
emprobforempty   0.4  (f-b-trn: probability for empty word)

// parameters modifying the EM-algorithm:
// --------------------------------------
m5p0   -1
// (fixed value for parameter p_0 in IBM-5 (if negative then it
//  is determined in training))
manlexfactor1   0  ()
manlexfactor2   0  ()
manlexmaxmultiplicity   20  ()
maxfertility   10  (maximal fertility for fertility models)
p0   -1
// (fixed value for parameter p_0 in IBM-3/4 (if negative
// then it is determined in training))
pegging   0  (0: no pegging; 1: do pegging)
```

Computational Intelligence and Learning Doctoral School

U. C. Louvain

Machine Translation:

Finite-State Models and Statistical Approaches

Statistical Framework for Machine Translation

Enrique Vidal

Pattern Recognition and Human Language Technology Group

Instituto Tecnológico de Informática

Departamento de Sistemas Informáticos y Computación

Universisdad Politécnica de Valencia, Spain

September 2007

E. Vidal – ITI-UPV-DSIC

Index

Notation and Basic Concepts

- x and y will generally denote *source* and *target* texts, respectively
- UNCONDITIONAL, CONDITIONAL AND JOINT PROBABILITIES:
  $$\Pr(X = x),\ \Pr(X = x \mid Y = y),\ \Pr(X = x, Y = y)$$
  Notation: $\Pr(x), \qquad \Pr(x \mid y), \qquad \Pr(x, y)$
- BAYES' RULE: $\Pr(x, y) = \Pr(x \mid y) \cdot \Pr(y) = \Pr(y \mid x) \cdot \Pr(x)$
- CHAIN RULE:
  $\Pr(x_1, x_2, \ldots, x_I) = \Pr(x_1) \cdot \Pr(x_2 \mid x_1) \cdots \Pr(x_I \mid x_1, \ldots, x_{I-1})$
  Notation: $\Pr(x_1^I) = \Pr(x_1) \cdot \Pr(x_2 \mid x_1) \cdots \Pr(x_I \mid x_1^{I-1})$
- MARGINAL: $\Pr(x) = \sum_y \Pr(x, y)$
- MODE: $\hat{x} = \underset{x}{argmax} \Pr(x)$: $\Pr(\hat{x}) = \underset{x}{max} \Pr(x)$
- MODE APPROXIMATION: $\sum_x \Pr(x) \approx \underset{x}{max} \Pr(x)$

Index

General framework

- Every sentence y in one language is a possible translation of any sentence x in another language.
- For each possible pair of sentences, y and x, there is a probability $\Pr(y \mid x)$.
- The probability should be *low* for pairs of sentences such as:

  quiero una habitación doble con vistas al mar
  are all expenses included in the bill ?

- The probability should be *high* for pairs of sentences such as:

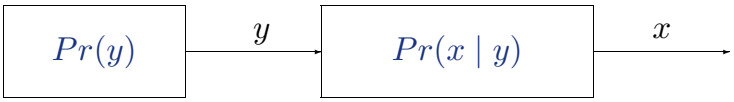
  ¿ hay alguna habitación tranquila libre ?
  is there a quiet room available ?

An inverse approach

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

$$\hat{y} = \underset{y}{argmax}\, \Pr(y \mid x) \;=\; \underset{y}{argmax}\, \frac{\Pr(x \mid y) \cdot \Pr(y)}{\Pr(x)} \;=\; \underset{y}{argmax}\, \Pr(x \mid y) \cdot \Pr(y)$$

A "distorted (noisy) channel model"



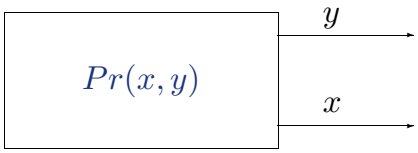
***Need:** a target-language model + alignment and lexicon models*

A finite-state approach

The direct probability can be decomposed in a different way:

$$\hat{y} = \underset{y}{argmax}\, \Pr(y \mid x) \;=\; \underset{y}{argmax}\, \frac{\Pr(x,y)}{\Pr(x)} \;=\; \underset{y}{argmax}\, \Pr(x,y)$$

A "joint" model



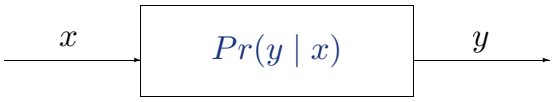
A stochastic finite-state transducer can model the joint distribution

A direct approach

Search for a target sentence with maximum *posterior* probability:

$$\hat{y} = \underset{y}{argmax}\, \Pr(y \mid x)$$

A "direct model"



Log-linear combination of models

A log-linear approach

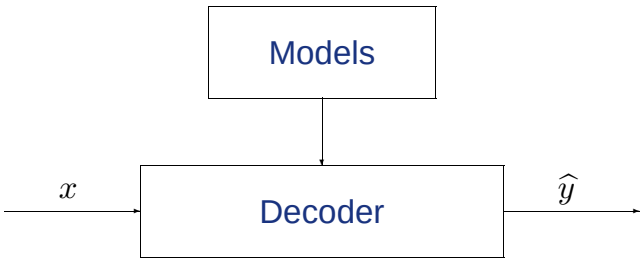
Search for a target sentence with maximum *posterior* probability:

$$\hat{y} = \underset{y}{argmax}\, \Pr(y \mid x)$$

$$\hat{y} = \underset{y}{argmax} \frac{\exp\left(\sum_{k=1}^{K} \lambda_k h_k(x,y)\right)}{\sum_{y'} \exp\left(\sum_{k=1}^{K} \lambda_k h_k(x,y')\right)} = \underset{y}{argmax} \sum_{k=1}^{K} \lambda_k h_k(x,y)$$

- $h_1(x,y) = \log Pr(y)$, a language model
- $h_2(x,y) = \log Pr(y \mid x)$, a translation model
- $h_3(x,y) = \log Pr(x \mid y)$, an inverse translation model
- $\ldots$

Translation search



$$\hat{y} = \underset{y}{argmax}\, Pr(y \mid x)$$

- Direct approach: log-linear combination of models (target-language model + translation models + $\ldots$)
- Inverse approach: target-language model + alignment and lexicon models
- Joint approach: stochastic finite-state transducer

Index

Bibliography

1. P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, P. Roosin: *A statistical approach to machine translation*. Computational Linguistics, 16:79–85, 1990.

2. F.J. Och, H. Ney: *Statistical Machine Translation*. EAMT Workshop, pp. 39-46, Ljubljana, Slovenia, May 2000.

3. F. Casacuberta, E. Vidal. *Machine translation with inferred stochastic finite-state transducers.* Computational Linguistics, 30(2):205–225. 2004.

4. F.J. Och: *Statistical Machine Translation: Foundations and Recent Advances.* Tutorial at MT Summit, Phuket, Thailand. 2005

Computational Intelligence and Learning Doctoral School
U. C. Louvain

Machine Translation:
Finite-State Models and Statistical Approaches
Statistical Alignment Models

Francisco Casacuberta and Enrique Vidal

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

September 2007

E.Vidal – ITI-UPV-DSIC

Index

General framework

- Every sentence y in one language is a possible translation of any sentence x in another language.
- For each possible pair of sentences, y and x, there is a probability $\Pr(y \mid x)$.
- The probability should be *low* for pairs of sentences such as:

  quiero una habitación doble con vistas al mar
  are all expenses included in the bill ?

- The probability should be *high* for pairs of sentences such as:

  ¿ hay alguna habitación tranquila libre ?
  is there a quiet room available ?

General framework

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

$$\hat{y} = \underset{y}{argmax} \Pr(y \mid x)$$

Approaches

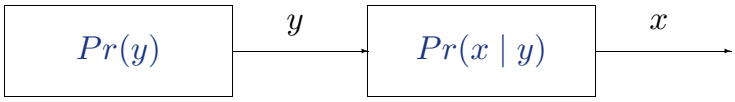
- An inverse approach: *channel models*
- A direct approach: *log-linear models*

The channel-source approach

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

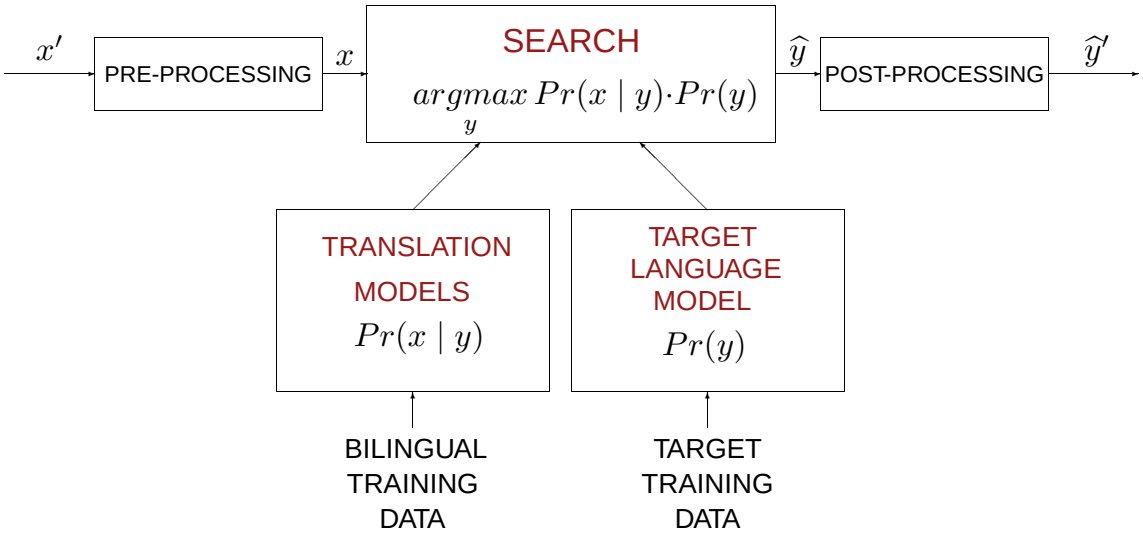
$$\hat{y} = \underset{y}{argmax}\, \Pr(y \mid x) = \underset{y}{argmax}\, \Pr(x \mid y) \cdot \Pr(y)$$

A channel model

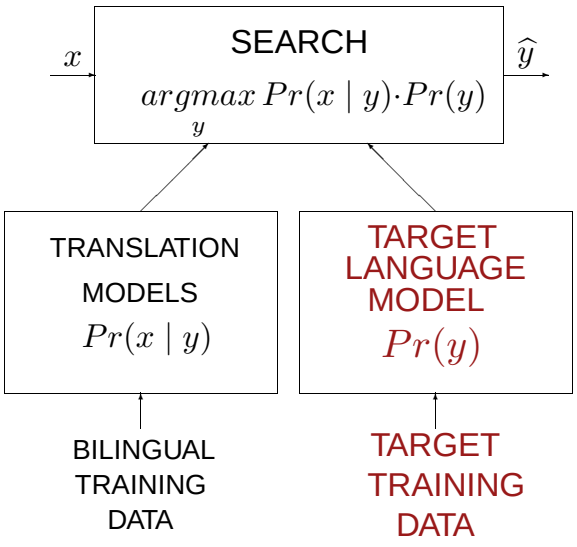


A target-language model + alignment and lexicon models

The channel-source approach



The target language model



Language models

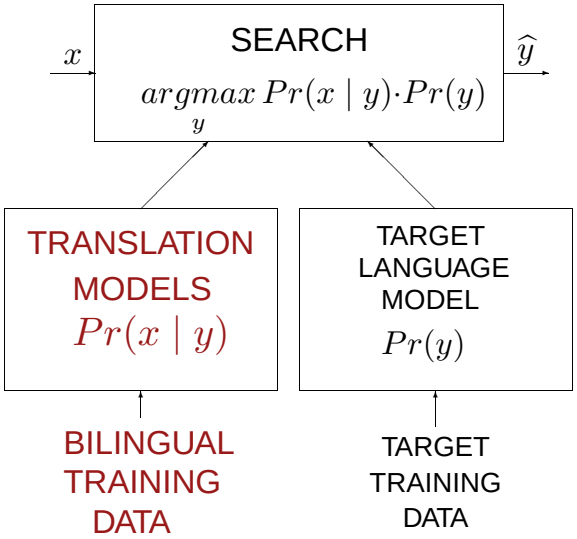
Word n-grams, word category N-grams, regular or context-free grammars, . . .

Language model learning:

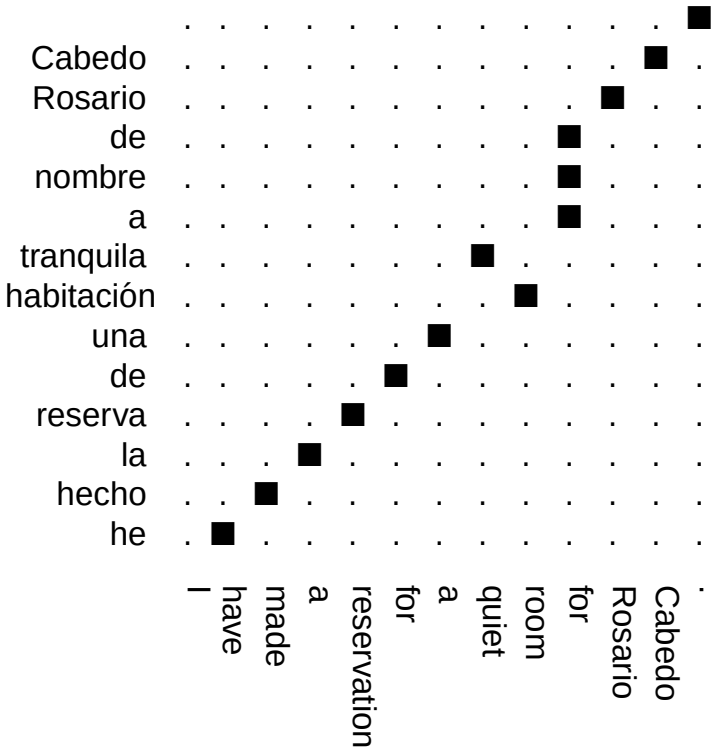
- Probabilistic estimation techniques.
- Grammatical inference techniques.
- SMOOTHING.
- Extensions: cache, triggers, categories, etc.
- Combinations: category n-grams and word m-grams, etc.
- Widely used toolkit for n-grams:
  - SRILM - The SRI Language Modeling Toolkit
    `http://www.speech.sri.com/projects/srilm/`

Index

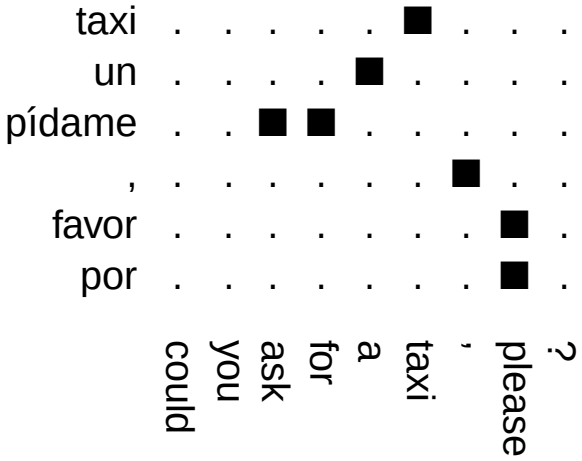
The translation model



Example of word alignments

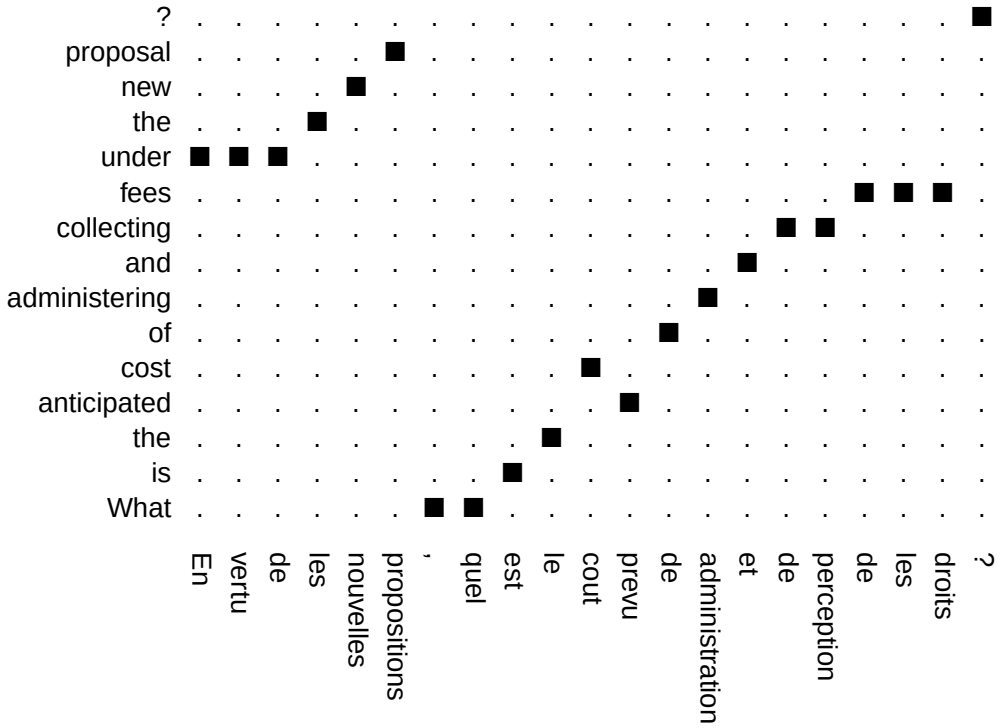


Example of word alignments



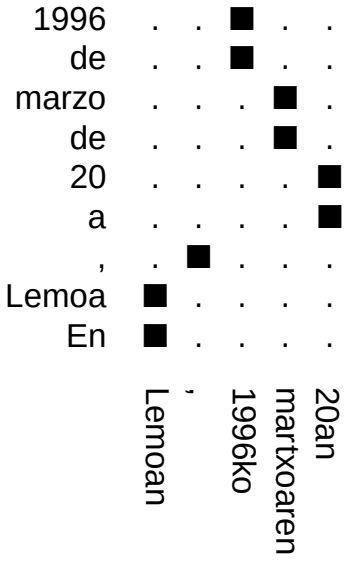
Example of word alignments

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



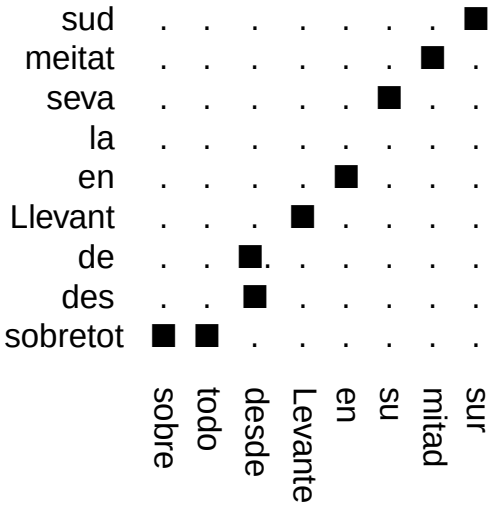
Example of word alignments

AMETRA corpus



Example of word alignments

METEO corpus



Alignments

- **Alignments** (Brown et al. 90):

  $a \subseteq \{1, ..., J\} \times \{1, ..., I\}$, $J =\mid x \mid, I =\mid y \mid$

  - Number of connections: $I \cdot J$; Number of alignments: $2^{I \cdot J}$
  - Constraint: $a : \{1, ..., J\} \rightarrow \{0, ..., I\}$,
    ($a_j = 0 \Rightarrow j$ in x is not aligned with any position in y).
  - Number of alignments: $(I + 1)^J$

- Set of possible alignments: $\mathcal{A}(x, y) = \{a : \{1, ..., J\} \rightarrow \{0, ..., I\}\}$

- Notation:

$$\begin{aligned} x &\equiv x_1, \ldots, x_J \equiv x_1^J \\ y &\equiv y_1, \ldots, y_I \equiv y_1^I \\ a &\equiv a_1, \ldots, a_J \equiv a_1^J \end{aligned}$$

Alignments

- Probability of translating y to x, $\Pr(x \mid y)$

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

$\Pr(x, a \mid y)$ is the probability of translating y to x through the alignment a

- Conditioning on source-sentence length, J:

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

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

- Length probability: $\Pr(J \mid y) \approx n(J|I)$

Alignments

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

- *Alignment probability:* $\Pr(a \mid J, y)$
- *Lexicon probability:* $\Pr(x \mid a, J, y)$

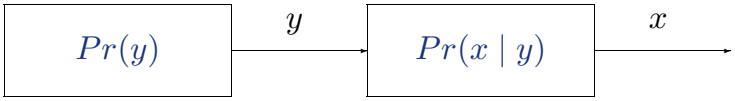
$$\Pr(a \mid J, y) = \prod_{j=1}^{J} \Pr(a_j \mid a_1^{j-1}, J, y), \qquad \Pr(x \mid a, J, y) = \prod_{j=1}^{J} \Pr(x_j \mid x_1^{j-1}, a, J, y)$$

$$\Pr(x, a \mid J, y) = \prod_{j=1}^{J} \Pr(a_j \mid a_1^{j-1}, J, y) \cdot \Pr(x_j \mid x_1^{j-1}, a, J, y)$$

$$\Pr(x \mid y) = \Pr(J \mid y) \cdot \sum_{a \in \mathcal{A}(x,y)} \prod_{j=1}^{J} \Pr(a_j \mid a_1^{j-1}, J, y) \cdot \Pr(x_j \mid x_1^{j-1}, a, J, y)$$

Generating $\mathbf{x}$ given $\mathbf{y}$: example

A channel model

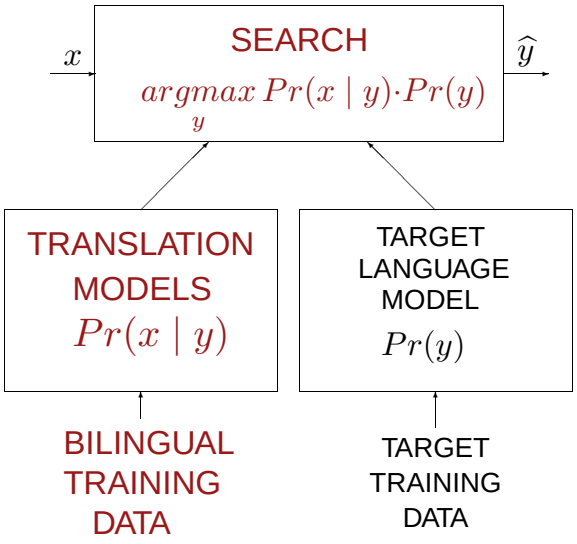


Given y = *a double room* $(I = 3)$

$\Pr(J \mid y) \rightarrow$ choose J: $J = 5$	1	2	3	4	5
$\Pr(a \mid J, y) \rightarrow$ choose a_j, $1 \leq j \leq J,\ 1 \leq a_j \leq 3$	1 a	3 room	2 double	2 double	2 double
$\Pr(x \mid a, J, y) \rightarrow$ choose x_j	Una	habitación	con	dos	camas

Index

The translation models



Zero-order translation models

- Model 1
- Model 2
- The Viterbi approximation
- The search problem

Model 1

$$\Pr(x, a \mid J, y) = \prod_{j=1}^{J} \Pr(a_j \mid a_1^{j-1}, J, y) \cdot \Pr(x_j \mid x_1^{j-1}, a, J, y)$$

- $\Pr(a_j \mid a_1^{j-1}, J, y) \approx \frac{1}{(I+1)}$
- $\Pr(x_j \mid x_1^{j-1}, a, J, y) \approx l(x_j \mid y_{a_j})$

- **Uniformly distributed alignmentes**
- **Statistical lexicon** defined by $l(u \mid v)$ where u, v are source and target vocabiulary words, respectively.

Model 1

$$\begin{aligned}
\Pr(x \mid y) &= \Pr(J \mid y) \cdot \sum_a \Pr(x, a \mid J, y) \\
&\approx n(J|I) \cdot \sum_a \prod_{j=1}^{J} \left[\frac{1}{(I+1)} \cdot l(x_j \mid y_{a_j}) \right] \\
&= \frac{n(J|I)}{(I+1)^J} \sum_{a_1=0}^{I} \cdots \sum_{a_J=0}^{I} \prod_{j=1}^{J} l(x_j \mid y_{a_j}) \\
&= \frac{n(J|I)}{(I+1)^J} \prod_{j=1}^{J} \sum_{a_j=0}^{I} l(x_j \mid y_{a_j}) \\
&= \frac{n(J|I)}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} l(x_j \mid y_i) \quad = \quad P_{M1}(x \mid y)
\end{aligned}$$

Model 1: parameter estimation

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

$$\mathcal{L}'_A(l) = \prod_{k=1}^{K} P_{M1}(x^{(k)} \mid y^{(k)}) = \prod_{k=1}^{K} \frac{n(J^{(k)}|I^{(k)})}{(I^{(k)}+1)^{J^{(k)}}} \cdot \prod_{j=1}^{J^{(k)}} \sum_{i=0}^{I^{(k)}} l(x_j^{(k)} \mid y_i^{(k)})$$

- Or the log-likelihood:

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

- Procedure: Expectation-maximization or growth transformations

E-M estimation of Model 1 parameters

Parameters to be estimated: $l(u \mid v)$ in:

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

Expectation

$$Q(l(u \mid v); \bar{l}(u \mid v)) = \sum_{k=1}^{K} \sum_{j=1}^{J^{(k)}} \sum_{i=0}^{I^{(k)}} \frac{l(u \mid v)}{\sum_{u'} l(u' \mid v)} \cdot \log \bar{l}(u \mid v)$$

Maximization

$$\bar{l}(u \mid v)) = \frac{\sum_{k=1}^{K} c(u \mid v; x^{(k)}, y^{(k)})}{\sum_{u'} \sum_{k=1}^{K} c(u' \mid v; x^{(k)}, y^{(k)})}$$

with $c(u \mid v; x^{(k)}, y^{(k)}) = \dfrac{l(u \mid v) \cdot \#(v, y^{(k)}) \cdot \#(u, x^{(k)})}{\sum_{i=0}^{I^{(k)}} l(u \mid y_i^{(k)})}$

Parameter estimation in Model 1

- E-M PROPERTY:
  each iteration increases the likelihood of the training set:

- MODEL-1 PROPERTY:
  eventually an absolute maximum is achieved!

- COMPUTATIONAL COST OF ONE ITERATION
  ($I_M = max_k\, I^{(k)}$, $J_M = max_k\, J^{(k)}$):
  - time: $O(K \times (I_M + J_M))$
  - space: $O(|\Sigma| \times |\Delta|)$

- Public software for training Model 1:
  `http://www.fjoch.com/GIZA++.html`
  `http://code.google.com/p/giza-pp`

Model 2

$$\Pr(x, a \mid J, y) = \prod_{j=1}^{J} \Pr(a_j \mid a_1^{j-1}, J, y) \cdot \Pr(x_j \mid x_1^{j-1}, a, J, y)$$

- $\Pr(a_j \mid a_1^{j-1}, J, y) \approx a(a_j \mid j, J, I)$
- $\Pr(x_j \mid x_1^{j-1}, a, J, y) \approx l(x_j \mid y_{a_j})$

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

$l(u \mid v)$ defines a **statistical lexicon**

Model 2

$$
\begin{aligned}
\Pr(x \mid y) &= \Pr(J \mid y) \cdot \sum_{a} \Pr(x, a \mid J, y) \\
&\approx n(J \mid I) \cdot \sum_{a} \prod_{j=1}^{J} \left[a(a_j \mid j, J, I) \cdot l(x_j \mid y_{a_j}) \right] \\
&= n(J \mid I) \cdot \sum_{a_1=0}^{I} \cdots \sum_{a_J=0}^{I} \prod_{j=1}^{J} \left[a(a_j \mid j, J, I) \cdot l(x_j \mid y_{a_j}) \right] \\
&= n(J \mid I) \cdot \prod_{j=1}^{J} \sum_{a_j=0}^{I} a(a_j \mid j, J, I) \cdot l(x_j \mid y_{a_j}) \\
&= n(J \mid I) \cdot \prod_{j=1}^{J} \sum_{i=0}^{I} a(i \mid j, J, I) \cdot l(x_j \mid y_i) \quad = \quad P_{M2}(x \mid y)
\end{aligned}
$$

Model 2: parameter estimation

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

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

- Procedure: Expectation-maximization or growth transformations

Model 2: parameter estimation

- E-M PROPERTY:
  each iteration increases the likelihood of the training set.

  Eventually a local maximum is achieved.

- COMPUTATIONAL COST ($I_M = max_k\, I^{(k)}$, $J_M = max_k\, J^{(k)}$):
  - time: $O(K \times I_M \times J_M)$
  - space: $O((|\Sigma| \times |\Delta|) + I_M + J_M)$

- Public software for training Model 2:

  `http://www.fjoch.com/GIZA++.html`

  `http://code.google.com/p/giza-pp`

Optimal alignment with Model 2

Search for a "best" alignment from $\mathcal{A}(x, y)$ in general:

$$\begin{aligned}\Pr(x \mid y) &= \Pr(J \mid y) \cdot \sum_{a \in \mathcal{A}(x,y)} \Pr(x, a \mid J, y) \\ &\approx \Pr(J \mid y) \cdot \max_{a \in \mathcal{A}(y,s)} \Pr(x, a \mid J, y) \;=\; \widehat{\Pr}(x \mid y)\end{aligned}$$

Using Model 2:

$$\begin{aligned}\widehat{\Pr}(x \mid y) &= \Pr(J \mid y) \cdot \max_a \Pr(x, a \mid J, y) \\ &\approx n(J \mid I) \cdot \max_a \prod_{j=1}^{J} \left[a(a_j \mid j, J, I) \cdot l(x_j \mid y_{a_j})\right] \\ &= n(J \mid I) \cdot \prod_{j=1}^{J} \max_{0 \le i \le I} \left[a(i \mid j, J, I) \cdot l(x_j \mid y_i)\right] = \widehat{P}_{M2}(x \mid y)\end{aligned}$$

Optimal alignment with Model 2

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

$$A[j] := \underset{0 \leq i \leq I}{argmax}\, a(i \mid j, J, I) \cdot l(x_j \mid y_i)$$

End-for
Return: A

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

Public software for training Models 1 and 2 and for computing the optimal alignments:

`http://www.fjoch.com/GIZA++.html`

`http://code.google.com/p/giza-pp`

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

---

por favor , he hecho una reserva a nombre de Federico Redondo .
I (4) have (4) made (5) a (6) reservation (5) for (9) Federico (11) Redondo (12) . (0)

---

por favor , ¿ podr´ıa pedir nuestro taxi ?
could (5) you (4) ask (6) for (6) our (7) taxi (8) , (3) please (3) ? (9)

¿ les importar´ıa despertarnos mañana a las siete y cuarto , por favor ?
would (2) you (1) mind (3) waking (4) us (4) up (6) tomorrow (5) at (7) a (9) quarter (10) past (9) seven (8) , (13) please (13) ? (1)

---

me voy a ir el jueves tres de junio a la una y media de la tarde .
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)

Alternative training procedure: Viterbi estimation

- Training sample: $A = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(K)}, y^{(K)})\}$
- Function to be maximized: Viterbi score ($\widehat{P}_{M2}$)

$$\widehat{\mathcal{L}}_A(a, l) = \prod_{k=1}^{K} \widehat{P}_{M2}(x^{(k)} \mid y^{(k)})$$

- Procedure: Expectation-maximization or growth transformations

Viterbi estimation

- PROPERTY: each iteration increases the Viterbi score.

  Eventually a local maximum is achieved.

- COMPUTATIONAL COST ($I_M = max_k\, I^{(k)}$, $J_M = max_k\, J^{(k)}$)
  - time: $O(K \times I_M \times J_M)$
  - space: $O((|\Sigma| \times |\Delta|) + I_M \times J_M)$

The translation process: searching

$$\underset{y}{argmax}\, Pr(x \mid y) \cdot Pr(y)$$

A computational difficult problem

(K.Knight *Decoding complexity in word-replacement translation models.* Comp. Ling. 1999)

ALGORITHMIC SOLUTIONS:

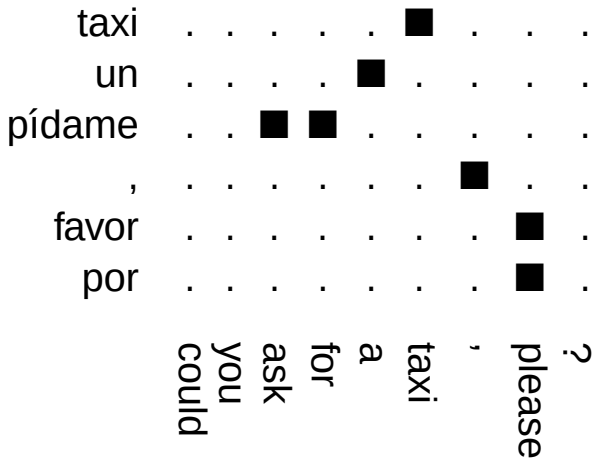
- Dynamic Programming like (Garcia-Varea, 1998) (Ney, 2000)
- Stack-Decoding A* or Branch & Bound (Brown, 1990) (Wang, 1997)
- Greedy (Germann, 2001)
- Using finite-state transducers (Kumar, 2004)

Other alignment models

- Models 2, 3, 4 and 5
- First-order (hidden Markov) models
- Recursive models
- Max-entropy models and log-linear combinations
- ***Phrase-based models and Alignment Templates***

Index

Exemple of word alignment

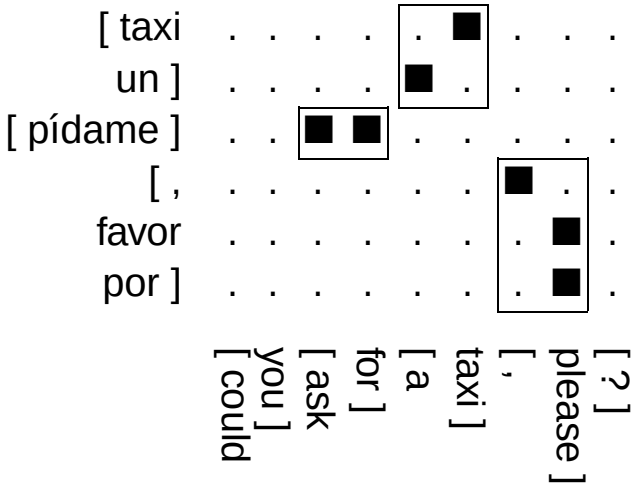


Segment alignment

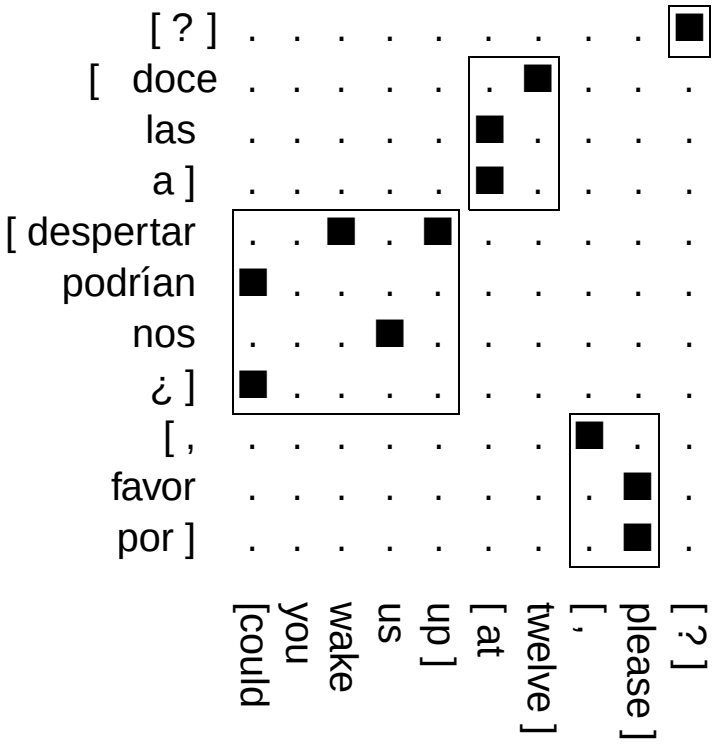
SINGLE-WORD ALIGNMENTS: only model the correspondence between words.

Alternative:

SEGMENT ALIGNMENTS: modelling the correspondences between word segments.



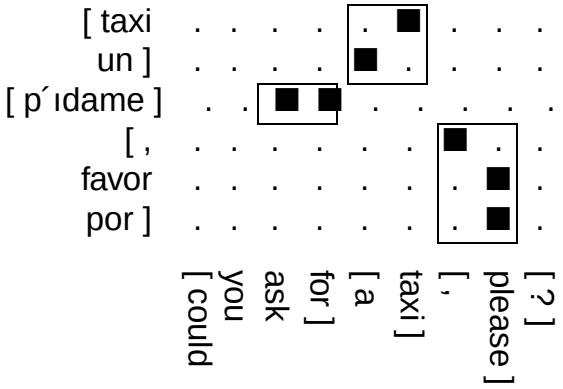
Segment alignment



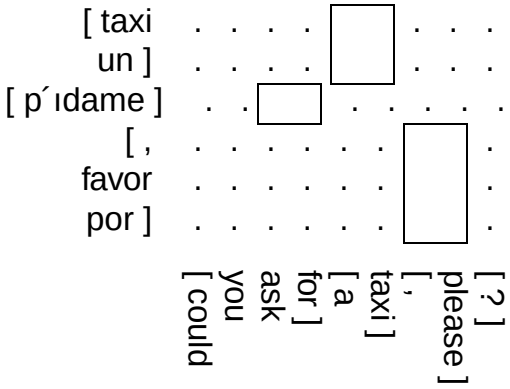
Beyond word-based models

- The basic assumption in the current word-based models: Each source word is generated by only one target word.
- This assumption does not naturally hold in natural language. In some cases, it is necessary to know the context.
- Solutions:
  - *Context-dependent dictionaries* 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.

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

An example

y = could you ask for a taxi , please ?

	y	could	you	ask	for	a	taxi	,	please	?	
	i	1	2	3	4	5	6	7	8	9=I	
Segmentation	μ				μ_1		μ_2			μ_3	
Permutation	α		$\alpha_1 = 2$			$\alpha_2 = 3$			$\alpha_3 = 1$		
		,	please		?	could you ask for			a	taxi	
Translation	x	por	favor		,	p´ıdame			un	taxi	.
	j	1	2		3	4			5	6	7
Segmentation	γ				γ_{α_3}	γ_{α_1}				γ_{α_2}	

x = por favor , p´ıdame un taxi .

Phrases and bilingual segmentation

- Let K be the number of segments.
- Segmentation of the target sentence

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

- Segmentation of the source sentence

$$\gamma : \{1, \ldots, K\} \rightarrow \{1, \ldots, J\} : \gamma_k \geq \gamma_{k-1} \;\; 1 < k \leq K \;\; \& \;\; \gamma_K = J \;\; (\gamma_0 = 0)$$

- Segment alignment (Permutation):

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

Monotone vs. no monotone alignments

NO MONOTONE ALIGNMENT

$$\Pr(x \mid y) \approx P(x \mid y) = p(J \mid I) \cdot \sum_{K} \sum_{\mu_1^K} \sum_{\alpha_1^K} \sum_{\gamma_1^K} \prod_{k=1}^{K} p(\alpha_k \mid \alpha_{k-1}) \cdot p(x_{\gamma_{\alpha_k-1}+1}^{\gamma_{\alpha_k}} \mid y_{\mu_{k-1}+1}^{\mu_k})$$

MONOTONE ALIGNMENT $\Rightarrow$ $\alpha_k = k$

$$\Pr(x \mid y) \approx P(x \mid y) = p(J \mid I) \cdot \sum_{K} \sum_{\mu_1^K} \sum_{\gamma_1^K} \prod_{k=1}^{K} p(x_{\gamma_{k-1}+1}^{\gamma_k} \mid y_{\mu_{k-1}+1}^{\mu_k})$$

Mode approximations

NO MONOTONE ALIGNMENT

$$\Pr(x \mid y) \approx \widehat{P}(x \mid y) = p(J \mid I) \cdot \max_{K, \mu_1^K, \alpha_1^K, \gamma_1^K} \prod_{k=1}^{K} p(\alpha_k \mid \alpha_{k-1}) \cdot p(x_{\gamma_{\alpha_k-1}+1}^{\gamma_{\alpha_k}} \mid y_{\mu_{k-1}+1}^{\mu_k})$$

MONOTONE ALIGNMENT $\Rightarrow$ $\alpha_k = k$

$$\Pr(x \mid y) \approx \widehat{P}(x \mid y) = p(J \mid I) \cdot \max_{K, \mu_1^K, \gamma_1^K} \prod_{k=1}^{K} p(x_{\gamma_{k-1}+1}^{\gamma_k} \mid y_{\mu_{k-1}+1}^{\mu_k})$$

Learning phrase-based models

- Models
  - Learning monotone phrase-based models
  - Learning non-monotone phrase-based models
- Phrase-based units
  - Training with a sentence-aligned corpus.
  - Training with a word-aligned corpus.

Learning monotone phrase-based models with a word-aligned corpus

Given a sentence-aligned corpus $\mathcal{T}$,

- A word-aligned corpus is generated using the GIZA++ toolkit with $\mathcal{T}$

  `http://code.google.com/p/giza-pp`

- A set of bilingual word sequences from the word aligned corpus is extracted.
- phrase-model parameters are estimated from frequency counts.

Learning monotone phrase-based models

Extracting bilingual multiword sequences: examples

x:	configuration	program	
y:	programa	de	configuración
a:	2	0	1

- BP_1 = {configuration-configuración, program-programa}
- BP_2 = {configuration-configuración, program-programa, configuration-de configuración, program-programa de, configuration program-programa de configuración}
- BP_3 = {configuration program-programa de configuración}

Learning non-monotone phrase-based models

- Estimation procedures are similar to those for monotone models.
- For the distortion model, $p(\alpha_k \mid \alpha_{k-1})$:

$$p(\alpha_k \mid \alpha_{k-1}) = p_0^{|\gamma_{\alpha_k} - \gamma_{\alpha_{k-1}}|} ,$$

where p_0 is a parameter to be ajusted using a validation set*.

F.Och, H. Ney *The Alignment Template Approach to Statistical Machine Translation*. Computational Linguistics, 30(4), 2004.

Search algorithms for monotone phrase-based models

- The 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 search strategy is generally based on the *multi-stack-decoding algorithm*.

Search algorithms for non-monotone phrase-based models

- The procedure is quite simmilar 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 *multi-stack* implementation requieres a stack for each possible subset of source positions and consequently, the computational cost can be very high.

Index

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.

- Possition-independent Word error rate (PER):
  Similar to WER, but not taking into account words order.

- 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).

Experiments with statistical word alignment models

- EuTrans-I corpus
- Hansards corpus
- Translation models: IBM $1^5 2^5 3^5 4^5 5^5$
  (5 bootstrap training iterations with GIZA++)
- Language models: 3-grams + *Good Turing* smoothing
- Diferent search/decoding strategies

Task definitions and corpora

- 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 (35,000/35,590 words), 2-Gram PP = 8.6/5.2
- The HANSARD corpus
  - Proceedings of the Canadian parliament (French $\rightarrow$ English)
  - Vocabulary: 58.016 French words and 42.055 English words
  - Training: 128.000 pairs
  - Test: 500 sentences of 4, 6, 8, 10, 12 words
  - First results in (Brown et al. 1993)
    - 12 training iterations (1 IBM1 + 6 IBM2 + 1 IBM3 + 4 IBM5)
    - Language model: trigrams
    - Search: stack-decoding
    - 48% of sentences from 73 were successfully translated

Experimental results: Search and modelling errors

EUTRANS-I corpus (all errors in %):

Strategy	seconds	Search Errs	Model Err	WER	PER
DPSearch-M2	55.7	5.5	55.2	12.7	10.5
DPSearch-M4	69.5	12.2	45.1	10.2	9.4
StackDecoding-M4	87.1	18.4	44.1	14.2	11.1
GreedySearch-M3	18.7	61.3	20.5	24.8	18.6
GreedySearch-M4	165.9	53.0	23.3	20.0	16.2

HANSARDS corpus (all errors in %):

Strategy	seconds	Search Errs	Model Err	WER	PER
DPSearch-M2	102.9	2.6	81.2	50.5	46.8
StackDecoding-M4	163.1	12.0	78.6	54.2	51.3
GreedySearch-M3	17.0	15.0	75.0	55.9	51.0

Experiments with phrase-based models

Additional corpora:

- “El Periódico”
- Xerox printer manuals
- Bulletin of the European Union

Corpora

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

		Spanish	Catalan
Train:	Sentence pairs	643,961	
	Running words (K words)	7,180	7,435
	Vocabulary (K,words)	129	128
Test:	Sentence pairs	240	
	Running words	4,316	4,389

Corpora

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

		En	Sp	En	Ge	En	Fr
Train:	Sentence K pairs	56		49		53	
	Runnig K words	665	753	633	696	587	534
	Vocabulary (K words)	8	11	8	10	8	19
Test:	Sentence pairs	1,125		984		996	
	Running K words	8	10	11	12	12	12
	Test perplexity	48	33	51	87	73	52

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

		En	Sp	En	Ge	En	Fr
Train:	Sentence K pairs	214		223		215	
	Runnig M words	5.9	6.6	6.5	6.1	6.0	6.6
	Vocabulary (K words)	84	97	87	152	85	91
Test:	Sentence pairs	800		800		800	
	Running K words	2	25	22	21	22	24
	Test perplexity	47	39	47	71	48	38

Results: Impact of different system choices and paramenters

Increasing the training set size. “El Periódico” task.

Corpus size (K words)	5	10	20	40	80	160	320	640
WER (%)	20.3	17.3	15.2	13.4	12.4	11.7	11.1	10.7
Parameters (millions)	0.1	0.2	0.4	0.7	1.2	2.1	3.6	7.0

Increasing the maximum segments length (MSL). “El Periódico” task.

MSL (words)	2	3	4
WER (%)	12.1	10.7	10.5
Parameters (millions)	2.0	7.0	14,5

Monotone vs. non monotone search. WER (%) for the XRCE task.

Search	En-Es	Es-En	En-Fr	Fr-En	En-De	De-En
Monotone	28.5	30.9	51.4	51.6	66.4	54.1
Non monotone	28.0	31.6	52.0	51.3	66.4	54.0

Comparison with other machine translation systems.

“El Periódico” task.

- **Salt** (`www.cultgva.es`), a knowledge-based machine translation systems supported by the Government of the Generalitat Valenciana.
- **Incyta** (`www.incyta.com`), a knowledge-based commercial system.
- **InterNOSTRUM** (`www.internostrum.com`), a hybrid knowledge-based and finite-state translation system.

Results

MT system	WER (%)	mWER (%)	BLEU
Salt	9.9	6.6	0.866
Incyta	10.0	7.6	0.855
InterNOSTRUM	11.9	8.5	0.837
Phrase-based	10.7	7.8	0.857

Index

Bibliography

- P.F. Brown et al. *A statistical approach to machine translation*. Computational Linguistics, vol. 16, pp. 79–85, 1990.
- P.F. Brown et al. *The mathematics of statistical machine translation: parameter estimation.* Computational Linguistics, vol. 19 (2), 263–310, 1993.
- I. Garcia-Varea, F. Casacuberta. *An iterative, DP-based search algorithm for statistical machine translation.* Proceedings of the ICSLP. 1998.
- P. Koehn, K. Knight. *Feature-rich statistical translation of noun phrases*. 41nd Annual Meeting of the ACL Sapporo, JAPAN. 311-318, July, 2003.
- F. J. Och, H. Ney: *A Systematic Comparison of Various Statistical Alignment Models* Computational Linguistics, 29(1):19-51, 2003.
- F.J. Och, H. Ney: *The Alignment Template Approach to Statistical Machine Translation*. Computational Linguistics, 30(4), 2004.
- J. Tomás, F. Casacuberta: *Monotone statistical translation using word groups*. In Proc. of the Machine Translation Summit VIII, pages 357-361, 2001.
- J. Tomás, F. Casacuberta: *Combining phrase-based and template-based alignment models in statistical translation*. Proc. of the IbPRIA, 2003.

Computational Intelligence and Learning Doctoral School

U. C. Louvain

Machine Translation:

Finite-State Models and Statistical Approaches

Finite-State Translation

Enrique Vidal

Pattern Recognition and Human Language Technology Group

Instituto Tecnológico de Informática

Departamento de Sistemas Informáticos y Computación

Universisdad Politécnica de Valencia, Spain

September 2007

E.Vidal – ITI-UPV-DSIC

Index

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 paradigm of *Interpretation*
- Interpretation tasks can 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:

$$\hat{y} \;=\; \underset{y}{argmax}\ \Pr(y \mid x) \;=\; \underset{y}{argmax}\ \Pr(x, y)$$

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

$$\Pr(x, y) \;\approx\; P_{\mathcal{T}}(x, y)$$

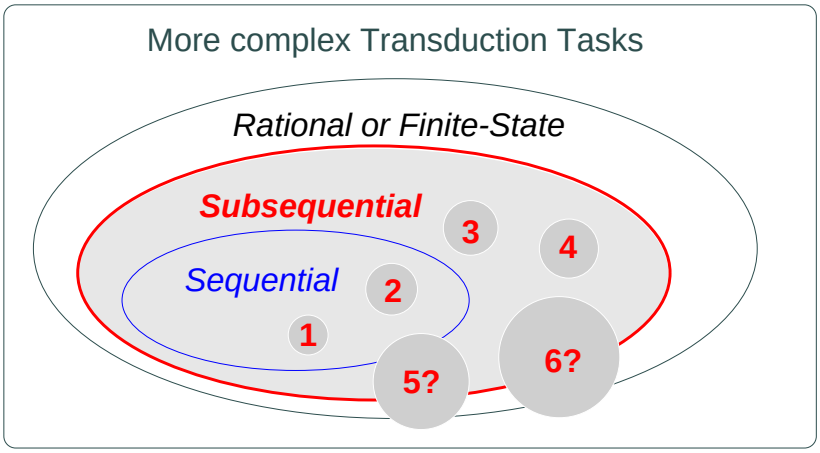
However, not all the transduction tasks are equally difficult. . .

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?
  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**
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE
  9 19 0 09
- **4... Roman to Decimal**
  III XIX XLII LXXIV CDII CMLXXXIX
  3 19 4 2 74 4 02 9 8 9
- **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

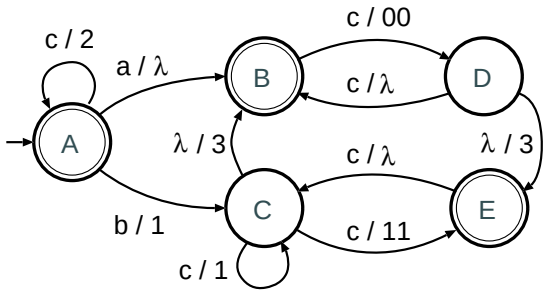
Finite State Transducers (FST): formal definition

A *Finite State* or *Rational Transducer* τ is a 6-tuple $\tau = (Q, X, Y, q_0, Q_F, E)$:

Q:	Finite set of *States*
X, Y:	Input and output *Alphabets*
$q_0 \in Q$:	*Initial State*
$Q_F \subset Q$:	Set of *Final States*
$E \subset Q \times X^* \times Y^* \times Q$:	*"Edges"* or *Transitions*

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

EXAMPLE

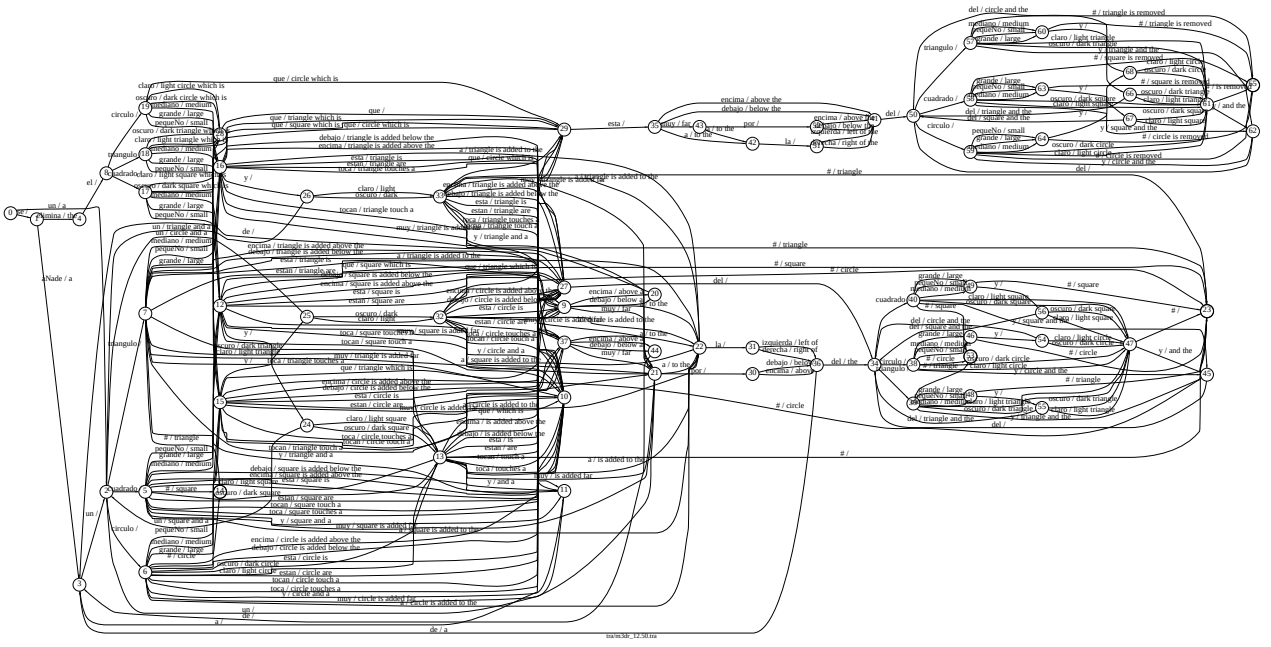


$T_\tau = \{$ (λ, λ), $(cb, 213)$, $(ccb, 2213)$,
(a, λ), $(ac, 003)$, $(cac, 2003)$,
$(c, 2)$, $(bc, 111)$, $(cbc, 2111)$,
$(b, 13)$, $(bc, 113)$, $(cbc, 213003)$,
$(ca, 2)$, $(bc, 13003)$, $(bcc, 1113)$,
$(cc, 22)$, $(cca, 22)$, $\ldots$ $\ldots$ $\}$

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*** $\mathcal{P}$ of a Finite State transducer $\tau = (Q, X, Y, q_0, Q_F, E)$ is a sequence of transitions of E
- A ***translation*** of τ is pair of strings $(x, y) \in X^* \times Y^*$ such that there is a *path* $\mathcal{P}$ in τ which "matches" x and y; that is:

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

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

- $T_\tau \subset X^* \times Y^* : T_\tau = \{(x, y)$ which are translations of $\tau\}$
- Let $\mathcal{P}(\tau, x, y)$ be the set of matching paths of x, y in τ.

  τ is ***ambiguous*** if $\exists x', y'$ such that $|\mathcal{P}(\tau, x', y')| > 1$

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

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 $y =$YES if x belongs to the language of G and $y =$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 $\forall x \in Dom(f), g(x) = f(x)$, when the number of pairs in S tends to infinity.

Stochastic Finite State Transducers

A *Stochastic* Finite State transducer $\mathcal{T}$ is defined by (τ, P, P_F), where:

- $\tau = (Q, X, Y, q_0, Q_F, E)$ is a Finite State transducer (with $Q_F = Q$)
- $P : E \rightarrow \mathbb{R}^+$ 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*, $\mathcal{P}_m$, ending at the state q_m:

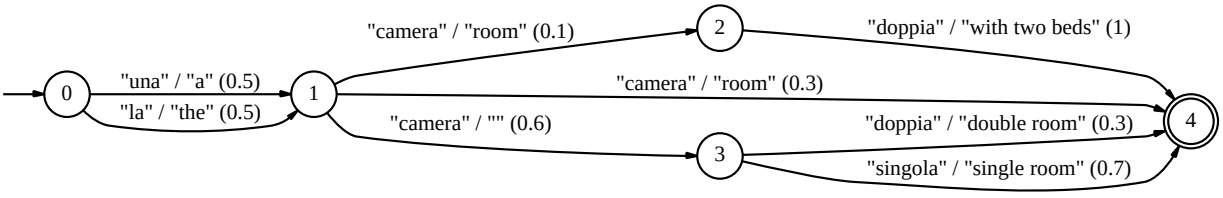
$$Pr(\mathcal{P}_m) \; = \prod_{(q',u,v,q)\in\mathcal{P}_m} P(q',u,v,q) \; P_F(q_m)$$

- Probability of a translation (x, y) of τ:

$$P_{\mathcal{T}}(x,y) \; = \sum_{\mathcal{P}_m\in\mathcal{P}(\tau,x,y)} Pr(\mathcal{P}_m) \; = \sum_{\mathcal{P}_m\in\mathcal{P}(\tau,x,y)} \; \prod_{(q',u,v,q)\in\mathcal{P}_m} P(q',u,v,q) \; P_F(q_m)$$

$P_{\mathcal{T}}(x,y)$ ***defines a joint distribution in*** X^*, Y^*

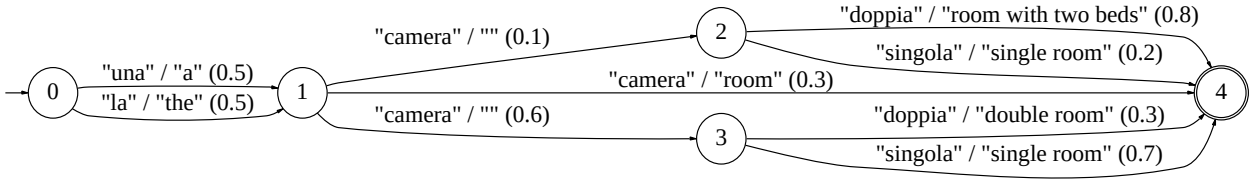
Example of a Stochastic Finite-State Transducer



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

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

Stochastic Finite-State Transducer: another example



$$Pr(\text{una camera singola , a single room}) =$$

$$0.5 \cdot 0.1 \cdot 0.2 + 0.5 \cdot 0.6 \cdot 0.7 = 0.01 + 0.21 = \mathbf{0.22}$$

Stochastic Finite State Transducers: embeeded language models

The marginals of the joint probability distribution $P_{\mathcal{T}}(x, y)$ defined by a stochastic finite-state transducer $\mathcal{T}$ are stochastic *regular* languages:

$$P_i(x) = \sum_{y \in Y^\star} P_{\mathcal{T}}(x, y), \quad P_o(y) = \sum_{x \in X^\star} P_{\mathcal{T}}(x, y).$$

These languages can be properly considered as *input* and *output Language Models* corresponding to $\mathcal{T}$.

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 $\mathcal{T}$, $x \in X^*$, $y \in Y^*$, find

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

Efficient solution by Dynamic Programming

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

$$\hat{y} = \underset{y \in Y^*}{argmax}\, P_{\mathcal{T}}(x, y)$$

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

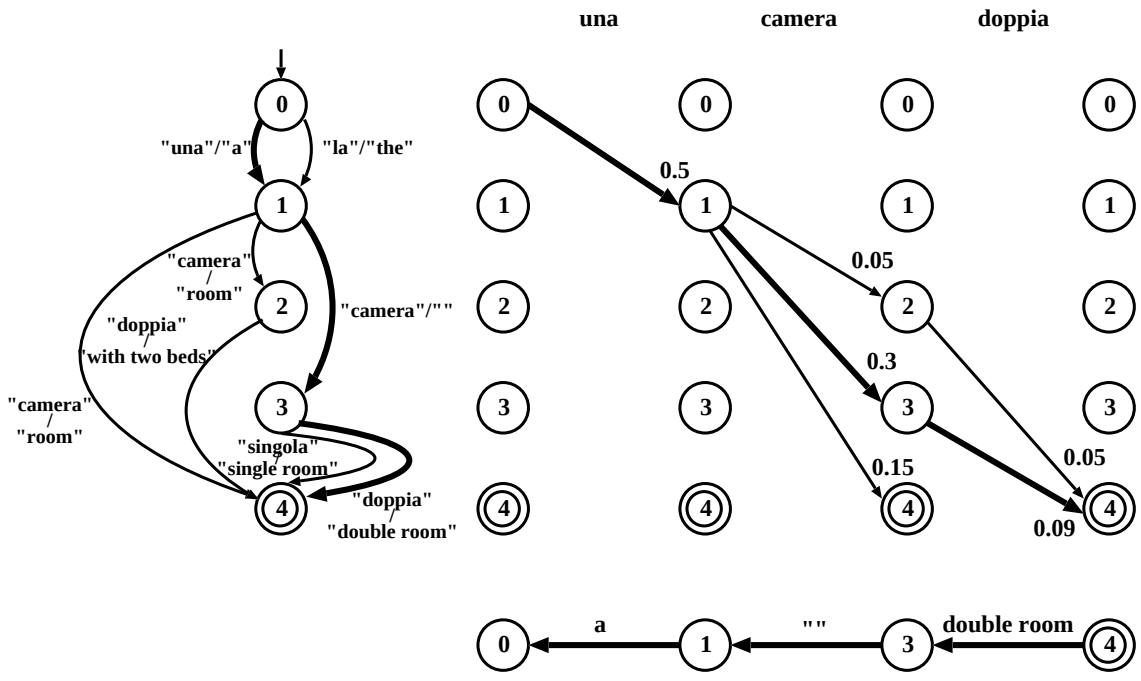
Approximation:

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

Efficient solution by Viterbi search

Both problems are easy if τ is un-ambiguous – trivial if τ is deterministic

Example of Viterbi translation



$$\underset{y}{argmax} \Pr(\text{“una camera doppia”}, y) \approx \text{“a double room”}$$

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!

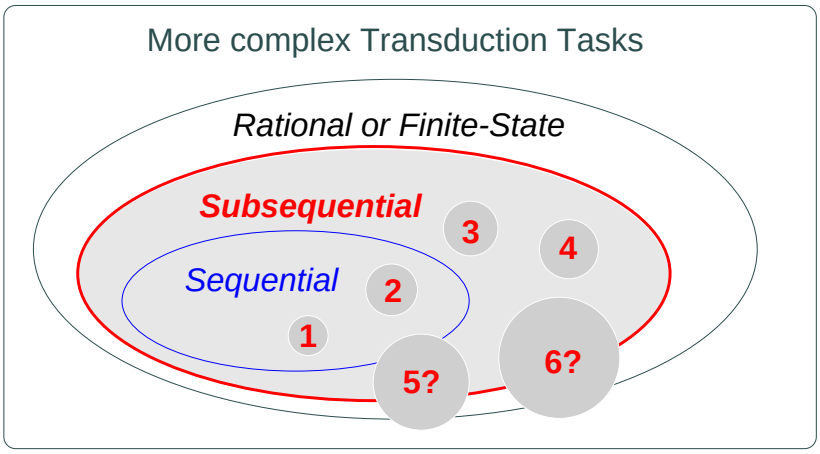
  $\Rightarrow$ Try to learn adequate subclasses and/or use heuristics!
- ***Hybrid methods***: Under the *traditional* paradigm, use statistical methods to guide the structure learning
- ***Pure statistical approach*** (*new*):
  - Adequately parameterize the SFST structure and consider it as a hidden 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

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*
$E \subset Q \times X \times Y^* \times Q$:	*"Edges"* or *Transitions*

- All the states are *accepting*
- Edges are *deterministic*:
  $(q, a, u, r), (q, a, v, s) \in E \Rightarrow (u = v \wedge r = s)$

PROPERTIES:

1. T_τ is a *function*: $X^* \rightarrow Y^*$
2. STs $\equiv$ *Generalized Sequential Machines* $\supset$ (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) τ 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 τ'; i.e.:

$$y = \tau(x) = \tau'(x)\sigma(q)$$

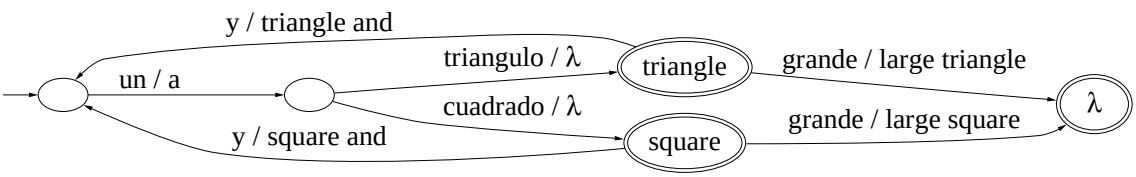
PROPERTIES:

1. T_τ 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:



Index

Learning SSTs: the OSTI Algorithm

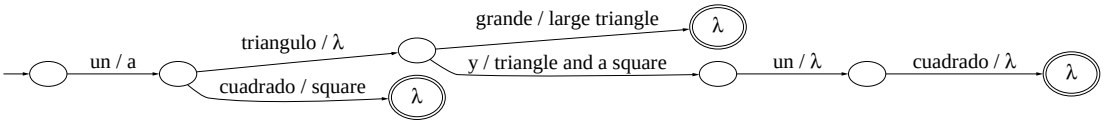
[Oncina, 91-93]

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.

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")

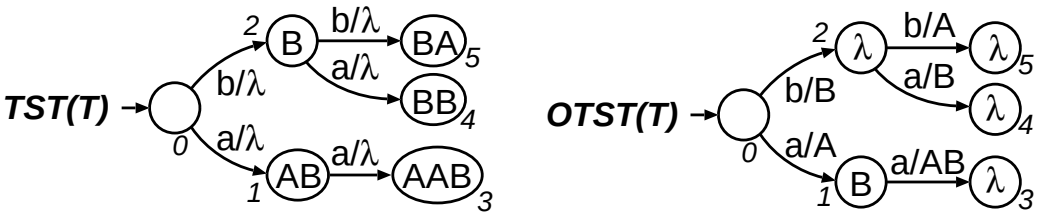
Input: Finite set of (non ambiguous) input output pairs $T \subset (X^* \times Y^*)$
Output: Onward Subsequential Transducer τ compatible with T

$\tau' = OTST(T)$; (let $Q(\tau')$ denote the set of states of τ')
$\forall\ q \in Q(\tau') - \{q_0\}$ in a *level-by-level order*, **do**
  $\forall\ p < q$ **do**
    $\tau = merge(\tau', p, q)$
    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 τ
      – **if** *"Derived Merging" possible* **then** $\tau = merge(\tau, q', q'')$
    end while
    if $subsequential(\tau)$ **then** $\tau' = \tau$
  end $\forall p$
end $\forall q$

end OSTIA

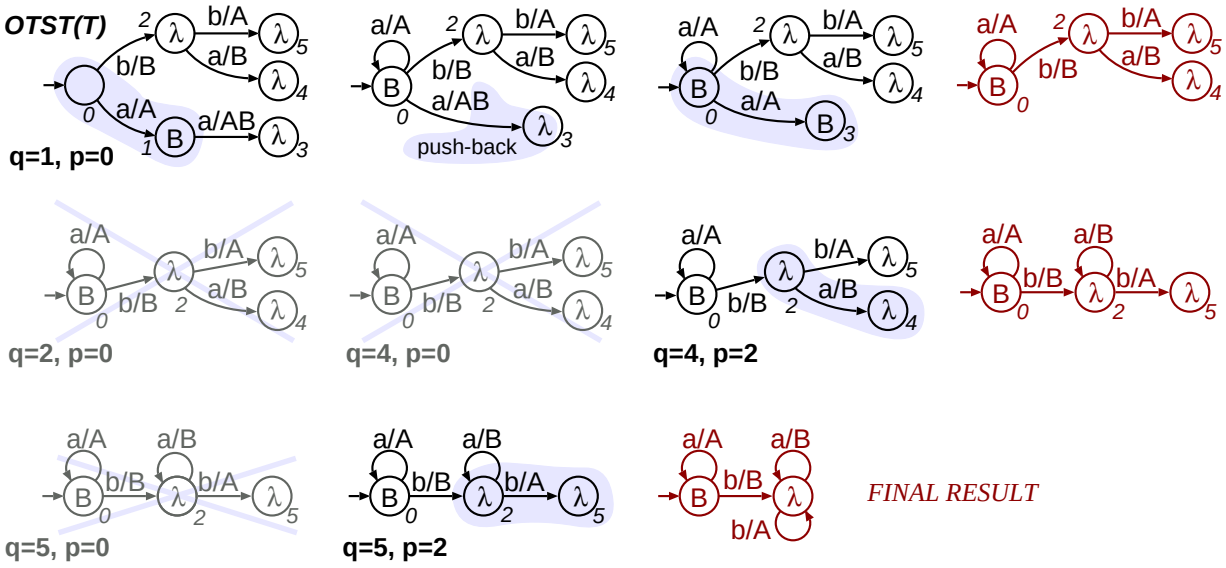
An Example of OSTIA state-merging process

X={a,b} ; 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 τ consistent with T

τ:=OTST(T); q:=$first(\tau)$;
while q $< last(\tau)$ **do** {
  q:=$next(\tau$,q); q':=$first(\tau)$;
  while q' $<$ q **do** {
    if σ(q') $= \sigma$(q) **or** σ(q') $= \emptyset$ **or** σ(q) $= \emptyset$ **then** {
      τ':=τ; $merge(\tau$,q',q);
      while $\neg subsequential(\tau)$ **do** {
        let (r,a,v,s), (r,a,v',s') be two edges of τ that
            violate the $subsequential$ condition, with s' $<$ s;
        if s' $<$ q **and** v' $\notin Pr$(v) **then exitwhile**;
        u:=lcp(v',v);
        $push\_back(\tau$,u^{-1}v', (r,a,v',s'));
        $push\_back(\tau$,u^{-1}v, (r,a,v,s));
        if σ(s') $= \sigma$(s) **or** σ(s') $= \emptyset$ **or** σ(s) $= \emptyset$
        then $merge(\tau$,s',s) **else exitwhile**;
      } // while $\neg subsequential(\tau)$
      if $\neg subsequential(\tau)$ **then** τ:=τ' **else exitwhile**;
    } // if σ(q') $= \sigma$(q)
    q':=$next(\tau$,q');
  } // while q' $<$ q
} // while q $< last(\tau)$

Properties of OSTIA learning

[Oncina, García & Vidal, 93]

- *Correctness*: the resulting transducer is *subsequential* and is a (state-merging) *generalization* of the set of training pairs T.
- *Convergence*: Using OSTIA the class of *total* Subsequential Transductions can be *identified in the limit*.
- *Efficiency*: OSTIA average running time is observed to be $O(n(m+k))$, where
  - $n = \sum_{(x,y)\in T} |x|$, (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]

Machine Translation (MT) and Subsequential Tansduction

- 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 asyncronies*
- ***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), winch 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.

Train. Size	WER	States	Edges
1,000	**58.8%**	412	1652
2,000	**57.0%**	846	3197
4,000	**51.8%**	1598	5970
8,000	**3.4%**	186	891
16,000	**0.0%**	17	206

- *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.

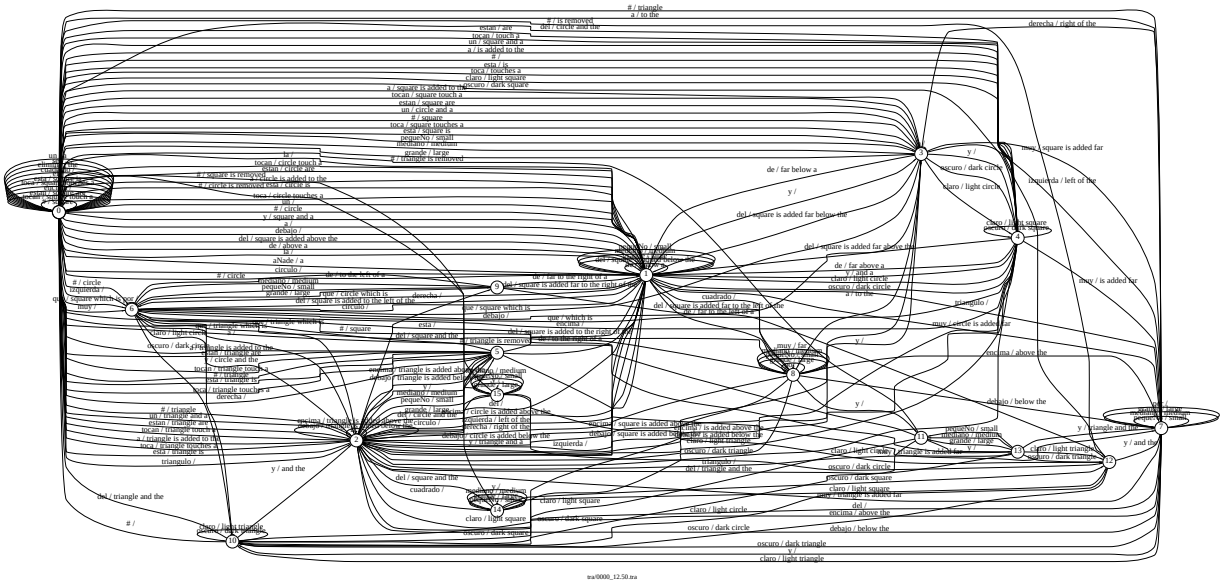
debajo izquierda esta por	→	square is removed
elimina un y	→	the a
a y y claro que	→	light square triangle which is
muy esta oscuro	→	dark square which is square

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*!

Basic OSTIA–learned SST for Spanish-English MTA

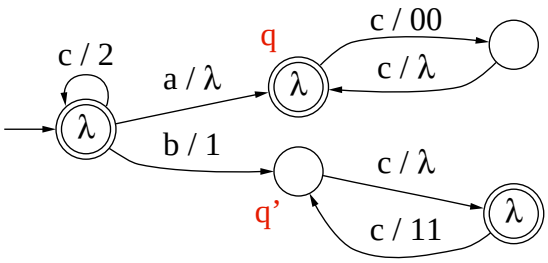


A Difficult-to-learn (partial) Subsequential Transduction

Let $t : \{a, b, c\}^* \rightarrow \{0, 1, 2\}^*$ be a *partial* Subsequential function defined as:

$$t = \{(c^m, 2^m) | m \geq 0\} \cup \{(c^m a c^{2n}, 2^m 0^{2n}) | m, n \geq 0\} \cup \{(c^m b c^{2n+1}, 2^m 1^{2n+1}) | m, n \geq 0\}$$

A Subsequential Transducer realizing t:

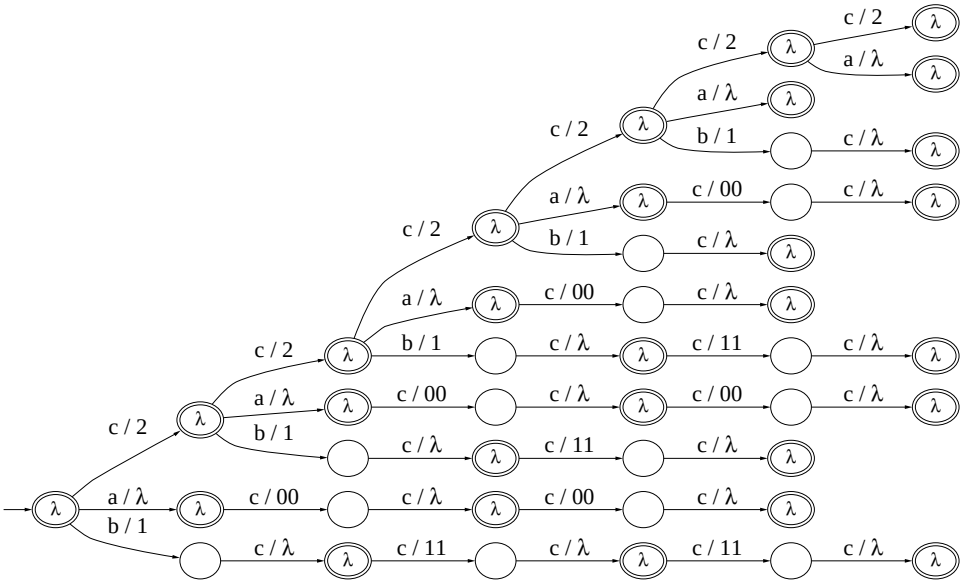


Samples of t, up to input length 6:

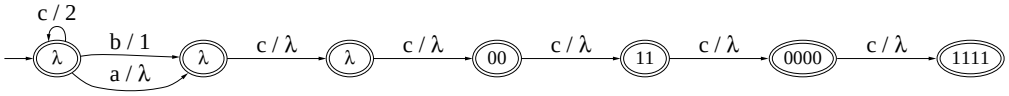
(,)	(*cbc*, 21)	(*cccc*, 2222)	(*bccccc*, 11111)
(*a*,)	(*cca*, 22)	(*acccc*, 0000)	(*cacccc*, 20000)
(*c*, 2)	(*ccc*, 222)	(*cbccc*, 2111)	(*ccbccc*, 22111)
(*bc*, 1)	(*bccc*, 111)	(*ccacc*, 2200)	(*cccacc*, 22200)
(*ca*, 2)	(*cacc*, 200)	(*cccbc*, 2221)	(*ccccbc*, 22221)
(*cc*, 22)	(*ccbc*, 221)	(*cccca*, 2222)	(*ccccca*, 22222)
(*acc*, 00)	(*ccca*, 222)	(*ccccc*, 22222)	(*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.

Helping OSTIA with input/output syntactic constraints

Two kind of conditions for OSTIA state merging:

- *Local conditions*: involve only the two states under consideration.

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

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

New Local Conditions:

Use *Finite-State Models* of the Input (or Domain) and/or the Output (or Range) to enforce *Input and/or Output Syntactic Constraints*

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

The resulting algorithm is known as OSTIA-DR

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]

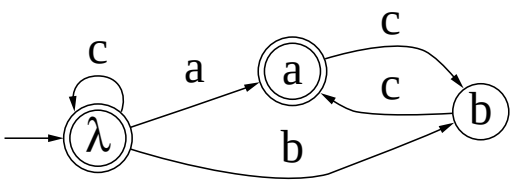
Algorithm OSTIA-DR ("OSTIA assisted by DOMAIN/RANGE constraints")
Input: Finite set of (non ambiguous) input output pairs $T \subset (X^* \times Y^*)$

Finite-State models, G_D, G_R, of the Domain (X^*) and Range (Y^*)

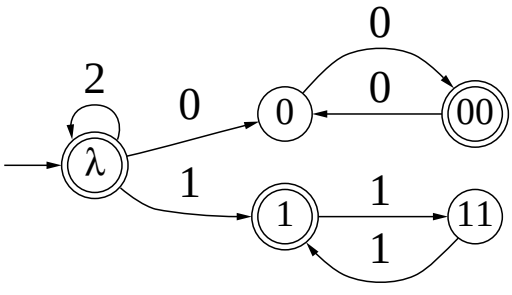
Output: Onward Subsequential Transducer τ' compatible with T
Method:

$\tau' = OTST(T)$; (let $Q(\tau')$ denote the set of states of τ')
$\forall\ q \in Q(\tau') - \{q_0\}$ in a *level-by-level order*, **do**
  $\forall\ p < q$ **if** p, q are compatible with G_D and/or G_R **do**
    $\tau = merge(\tau', p, q)$
    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 τ'
      – **if** *"Derived Merging" possible* **then** $\tau = merge(\tau, q', q'')$
    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"



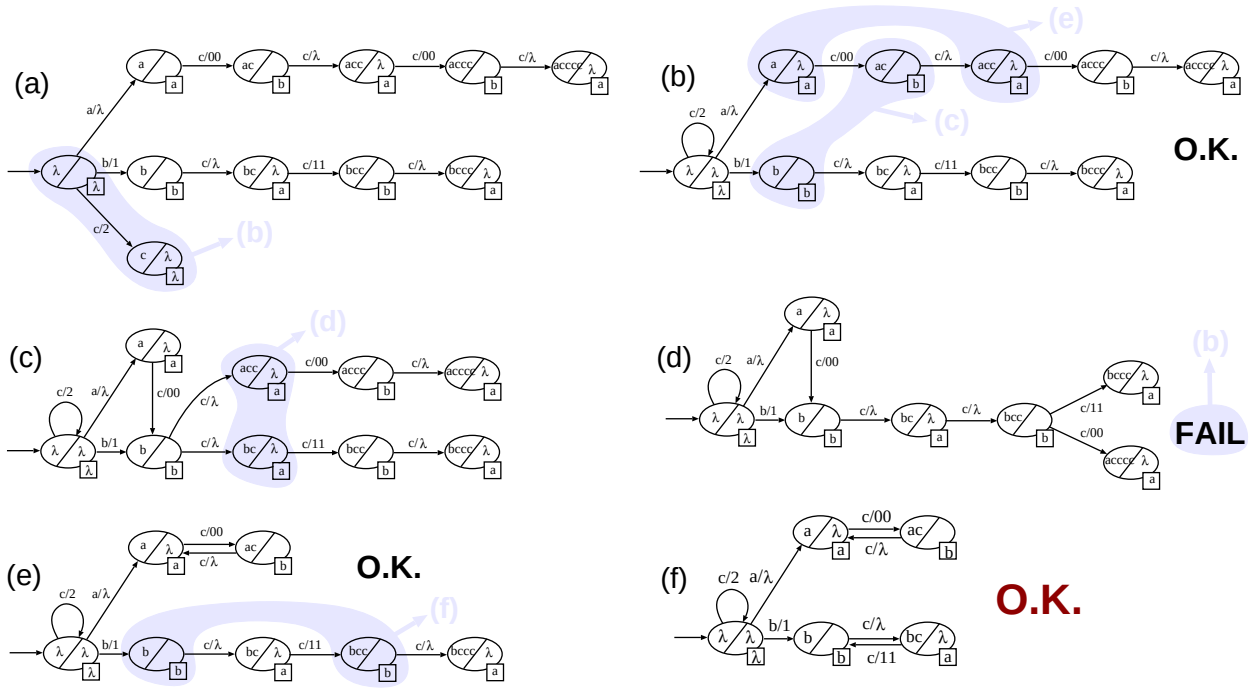
Finite State Domain (input) model



Finite State Range (output) model

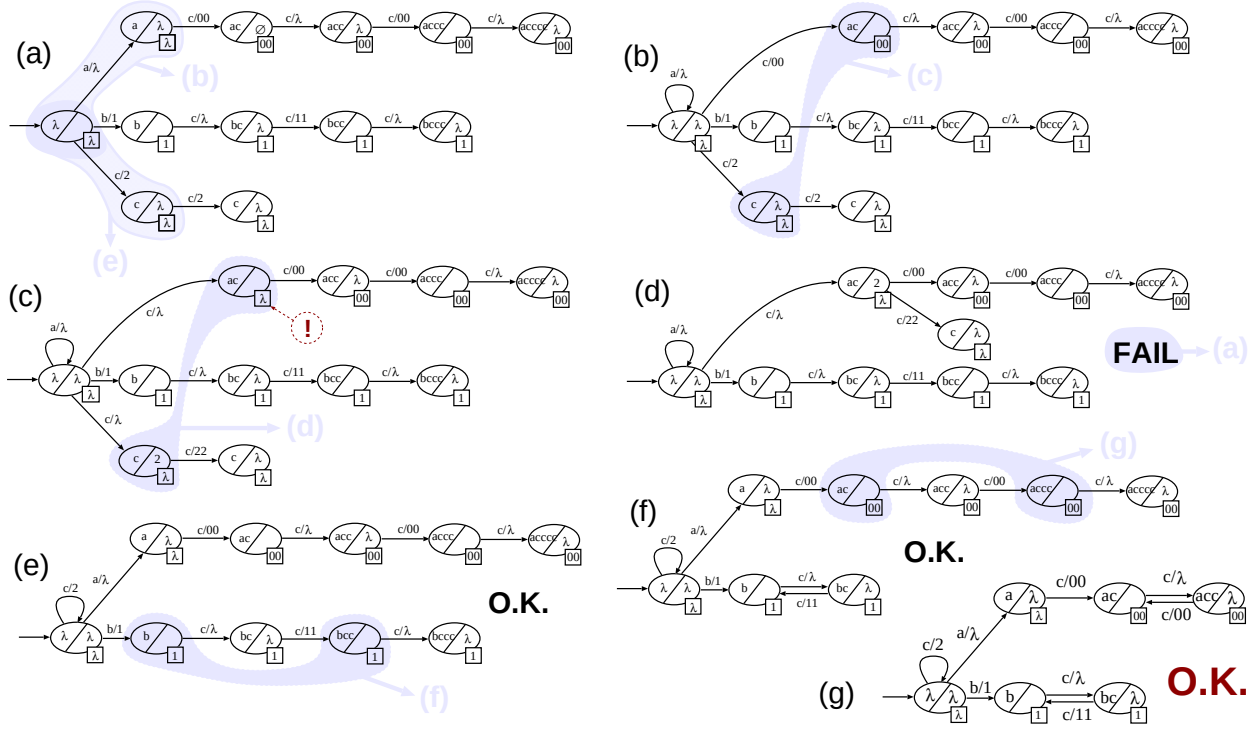
OSTIA-D learning

Training set: $T = \{(a, \lambda), (acc, 00), (acccc, 0000), (bc, 1), (bccc, 111), (c, 2)\}$



OSTIA-R learning

Training set: $T = \{(a, \lambda), (acc, 00), (acccc, 0000), (bc, 1), (bccc, 111), (c, 2), (cc, 22)\}$.



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 τ consistent with T and G_D

τ:=OTST(T); q:=$first(\tau)$;
while q $< last(\tau)$ {
  q:=$next(\tau$,q); q':=$first(\tau)$;
  while q' $<$ q {
    if (σ(q') $= \sigma$(q) **or** σ(q') $= \emptyset$ **or** σ(q) $= \emptyset$) **and**
     $\delta_D(p_0, input\_prefix(q')) = \delta_D(p_0, input\_prefix(q))$ **then** {
      τ':=τ; $merge(\tau$,q',q);
      while $\neg subsequential(\tau)$ {
        let (r,a,v,s), (r,a,v',s') be two edges of τ that
          violate the $subsequential$ condition, with s' $<$ s;
        if s' $<$ q **and** v' $\notin Pr$(v) **then exitwhile**
        u:=lcp(v',v);
        $push\_back(\tau$,u^{-1}v', (r,a,v',s')); $push\_back(\tau$,u^{-1}v, (r,a,v,s));
        if σ(s') $= \sigma$(s) **or** σ(s') $= \emptyset$ **or** σ(s) $= \emptyset$
        then $merge(\tau$,s',s) **else exitwhile**
      } // while $\neg subsequential(\tau)$
      if $\neg subsequential(\tau)$ **then** τ:=τ' **else exitwhile**
    } // if σ(q') $= \sigma$(q)
    q':=$next(\tau$,q');
  } // while q' $<$ q
} // while q $< last(\tau)$

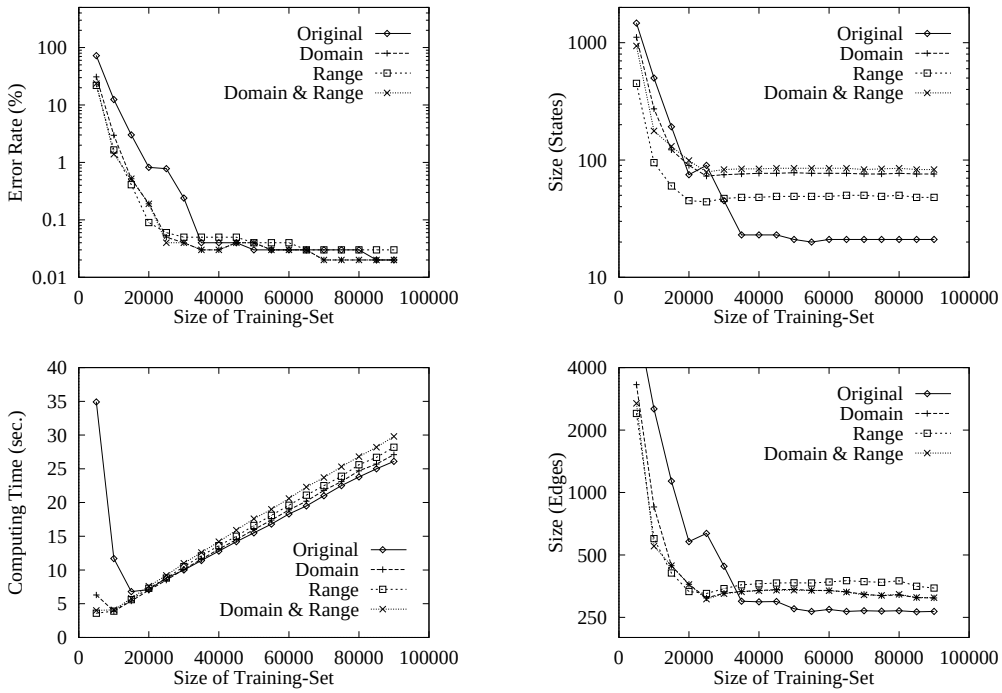
Using output-language constraints: OSTIA-R

INPUT: input-output pairs, $T \subset (X^* \times Y^*)$, Finite-State model, G_R, of the Range (X^*)
OUTPUT: OST τ consistent with T and G_R

τ:=OTST(T); q:=$first(\tau)$;
while q $< last(\tau)$ {
  q:=$next(\tau$,q); q':=$first(\tau)$;
  while q' $<$ q {
    if (σ(q') $= \sigma$(q) **or** σ(q') $= \emptyset$ **or** σ(q) $= \emptyset$) **and**
     $\delta_R(p_0, outut\_prefix(q')) = \delta_R(p_0, output\_prefix(q))$ **then** {
      τ':=τ; $merge(\tau$,q',q);
      while $\neg subsequential(\tau)$ {
        let (r,a,v,s), (r,a,v',s') be two edges of τ that
          violate the $subsequential$ condition, with s' $<$ s;
        if s' $<$ q **and** v' $\notin Pr$(v) **then exitwhile**
        u:=lcp(v',v);
        $push\_back(\tau$,u^{-1}v', (r,a,v',s')); $push\_back(\tau$,u^{-1}v, (r,a,v,s));
        if σ(s') $= \sigma$(s) **or** σ(s') $= \emptyset$ **or** σ(s) $= \emptyset$
        then $merge(\tau$,s',s) **else exitwhile**
      } // while $\neg subsequential(\tau)$
      if $\neg subsequential(\tau)$ **then** τ:=τ' **else exitwhile**
    } // if σ(q') $= \sigma$(q)
    q':=$next(\tau$,q');
  } // while q' $<$ q
} // while q $< last(\tau)$

MTA: OSTIA and OSTIA-DR learning performance

Spanish-English Extended MTA Learning performance as a function of training-set size. Domain and/or range Language Models: 3-TSS (3-Gram); Test Set: 100,000 independent input sentences.



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.

Training Set Size	OSTIA WER	OSTIA States	OSTIA Edges	OSTIA-DR WER	OSTIA-DR States	OSTIA-DR Edges
1,000	**58.8%**	412	1652	**55.1%**	813	2023
2,000	**57.0%**	846	3197	**47.1%**	1406	3353
4,000	**51.8%**	1598	5970	**30.1%**	1686	4051
8,000	**3.4%**	186	891	**1.4%**	244	719
16,000	**0.0%**	17	206	**0.0%**	100	363

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.

Train.Set Size	OSTIA Clean	OSTIA 5%Dist	OSTIA-DR Clean	OSTIA-DR 5%Dist
8,000	3.4%	**15.0%**	1.4%	**2.7%**
16,000	0.0%	**11.7%**	0.0%	**1.7%**

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:

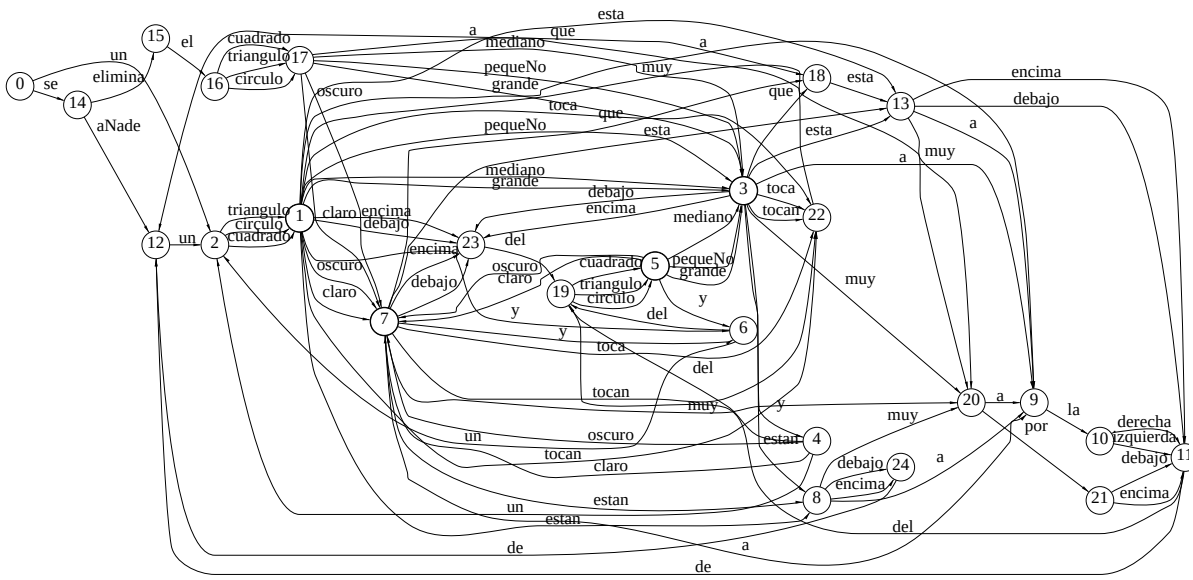
I : *se elimina* **el** *círculo grande y claro* **que** *está muy por encima* **del** *triángulo oscuro y del cuadrado* ● *mediano*
D: *se elimina* **y** *círculo grande y claro* ● *está muy por encima* ● *triángulo oscuro y del cuadrado* **un** *mediano*
T: the large light circle which is far above 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

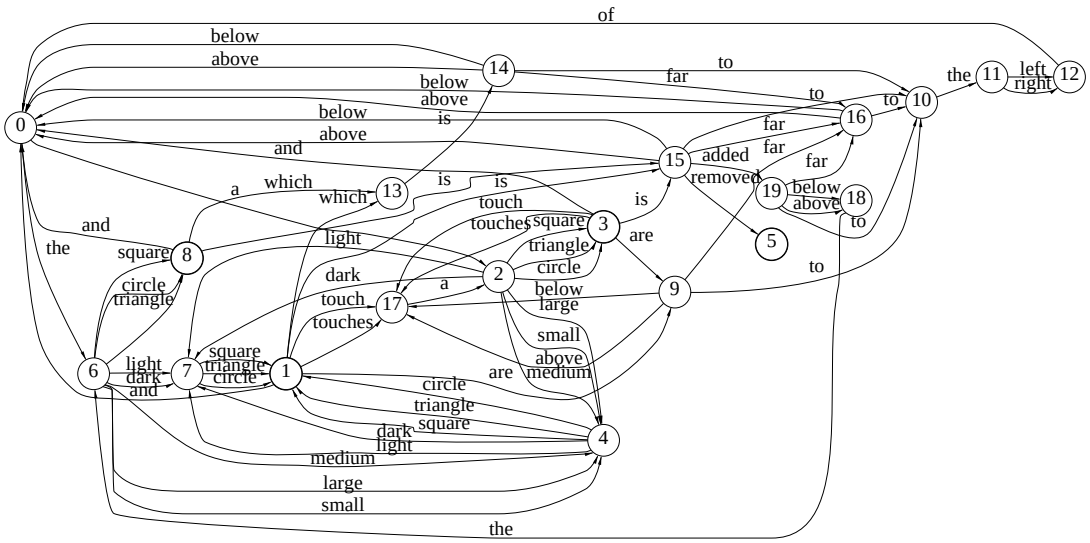
A Finite-State domain (Spanish) language model for MTA

3-TSS Automaton (entailing 3-Gram constraints)



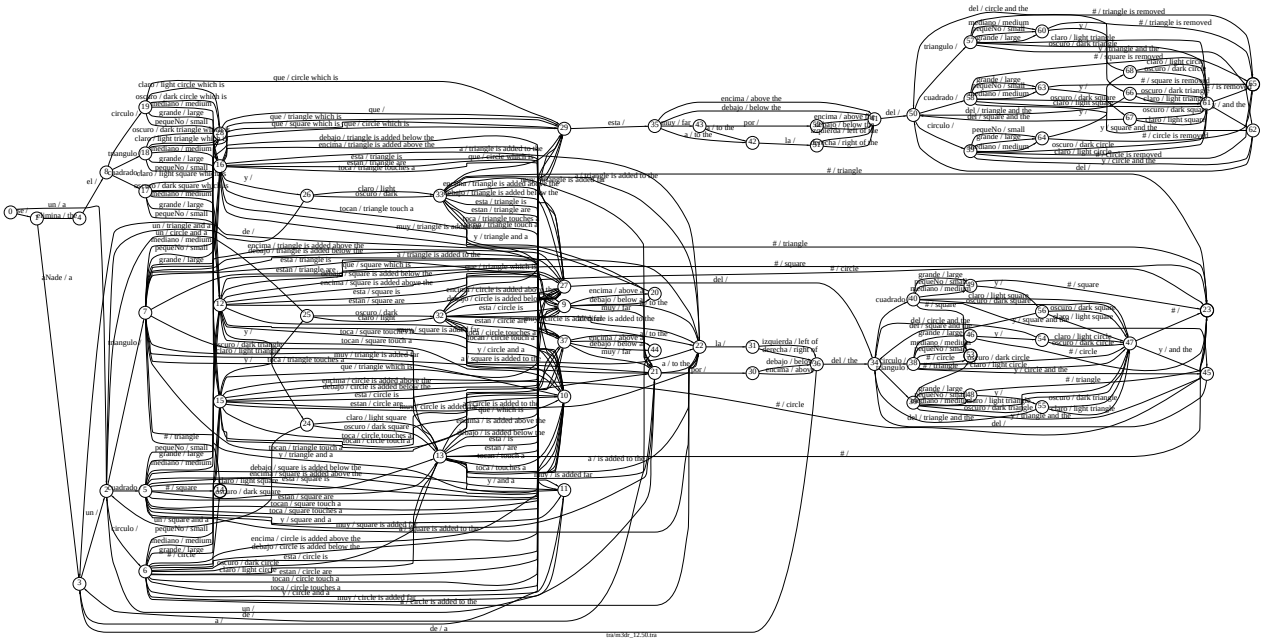
A Finite-State range (English) language model for MTA

3-TSS Automaton (entailing 3-Gram constraints)



OSTIA-DR–learned SST for Spanish-English MTA

(using both *Domain* and *Range* 3-Gram constraints)

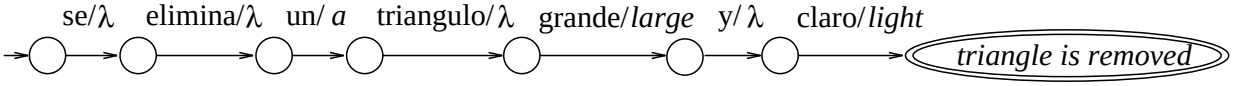


Scalability issues

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.

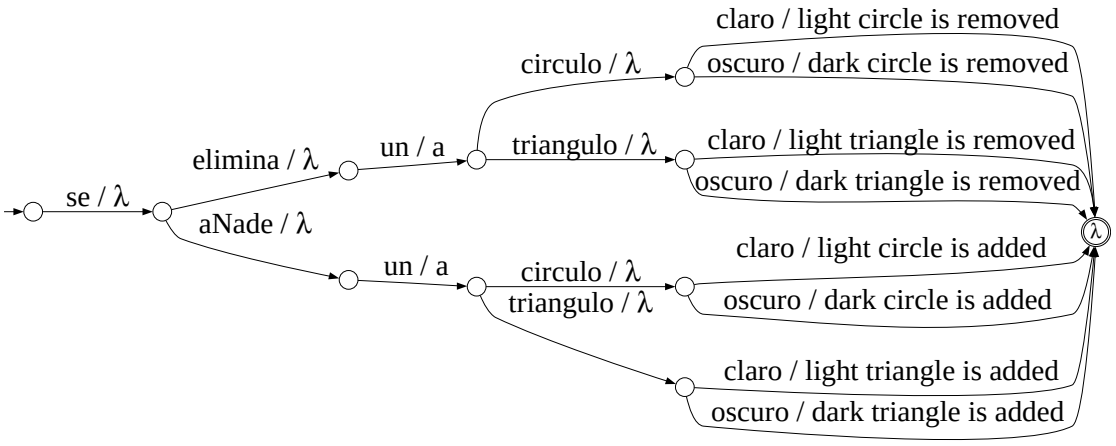
A very small MTA Spanish-English training set

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

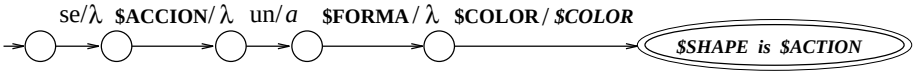
se $ACCION un $FORMA $COLOR ↔ a $COLOR $SHAPE is $ACTION

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 sahpes, 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.

Inp/Out	8,000 Train. Pairs		32,000 Train. Pairs	
Voc.Sizes	Direct	Categ.	Direct	Categ.
37/28	3.1	**0.9**	0.5	**0.2**
50/38	42.1	**1.5**	5.7	**0.3**
63/48	62.5	**3.0**	26.5	**0.6**
363/248	91.3	**3.4**	98.0	**0.7**

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.

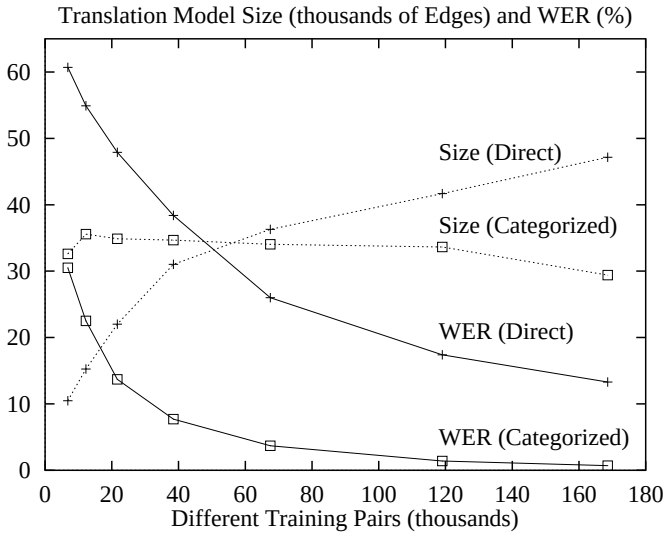
Por favor, prepárenos nuestra cuenta de la habitación dos veintidós.
Could you prepare our bill for room number two two two for us, please?

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.

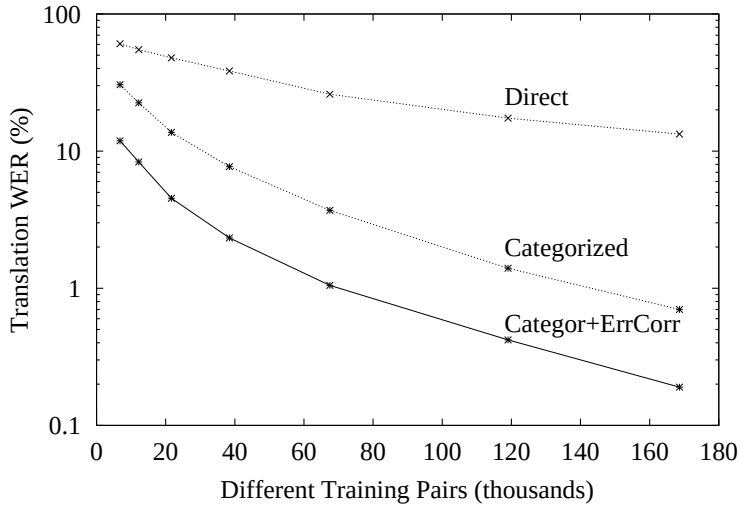
Test-Set:
2,730 different sentences.



▷ *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.



▷ ***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

	OSTIA-DR (baseline)	OSTIA-DR + Categories	OSTIA-DR + Categories+ECP
Spanish-English	13.33 %	0.74 %	0.18 %
Spanish-German	29.86 %	1.23 %	0.54 %
Spanish-Italian	17.60 %	2.54 %	0.51 %

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.

	Spanish-to-German	**Spanish-to-English**		
	EUTRANS	EUTRANS	Power Translator	Spanish Assistant
PCT	81.7%	87.3%	49.0%	
PCIT	93.3%	90.3%	79.7%	75.3%
UM	+0.86	+0.81	+0.64	+0.57

- **PCT**: Percentage of correct translations
- **PCIT**: Ppercentage 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:

$n \Rightarrow$ **Bilingual Categorization**
[Vilar, Marzal, Vidal, Eurospeech-95]:

While the direct approach degrades rapidly with increasing vocabulary sizes, categorization largely prevents accuracy degradation.

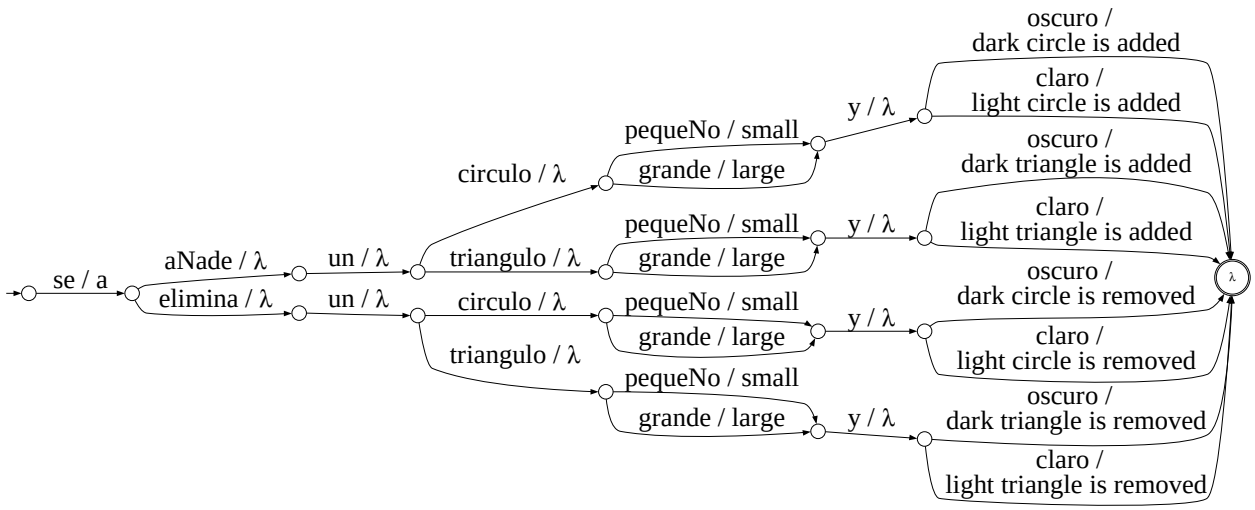
$k \Rightarrow$ ***Partial Alignment and Word Reordering***
[Vilar, Vidal, Amengual, Llorens, ECAI-96, SPECOM-96]:

Training-data requirements can be reduced dramatically.

A small training set from the MTA task

se elimina un triángulo grande y claro	↔	a large light triangle is removed
se elimina un triángulo pequeño y claro	↔	a small light triangle is removed
se elimina un círculo grande y claro	↔	a large light circle is removed
se elimina un círculo pequeño y claro	↔	a small light circle is removed
se elimina un triángulo grande y oscuro	↔	a large dark triangle is removed
se elimina un triángulo pequeño y oscuro	↔	a small dark triangle is removed
se elimina un círculo grande y oscuro	↔	a large dark circle is removed
se elimina un círculo pequeño y oscuro	↔	a small dark circle is removed
se añade un triángulo grande y claro	↔	a large light triangle is added
se añade un triángulo pequeño y claro	↔	a small light triangle is added
se añade un círculo grande y claro	↔	a large light circle is added
se añade un círculo pequeño y claro	↔	a small light circle is added
se añade un triángulo grande y oscuro	↔	a large dark triangle is added
se añade un triángulo pequeño y oscuro	↔	a small dark triangle is added
se añade un círculo grande y oscuro	↔	a large dark circle is added
se añade un círculo pequeño y oscuro	↔	a small dark circle is added

Transducer for the small MTA training set

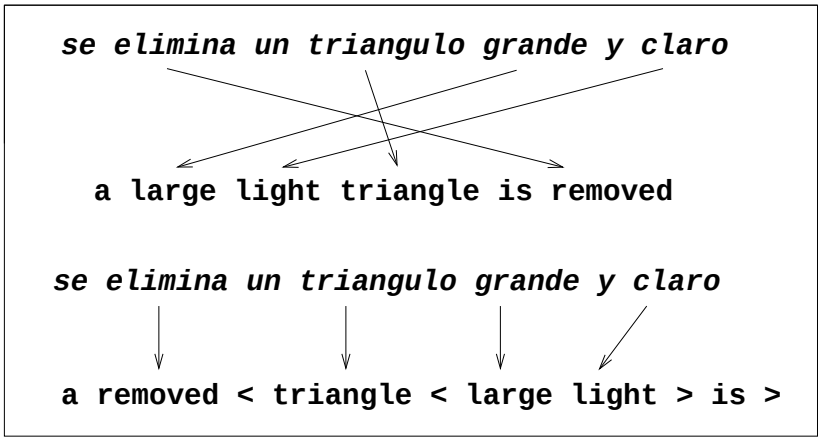


*Size **grows exponentially** with the number of words to be delayed.*

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*, wich can be obtained, e.g., using a probabilistic bilingual dictionary such as the one obtained by training an IBM-1 translation model.

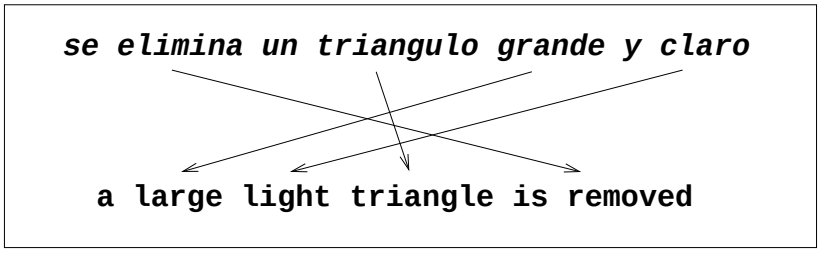


Original pair and a partial alignment (above). Reordering-Bracketing results (below).

Reordering is performed along with a *bracketing* scheme which allows recovering the correct word order.

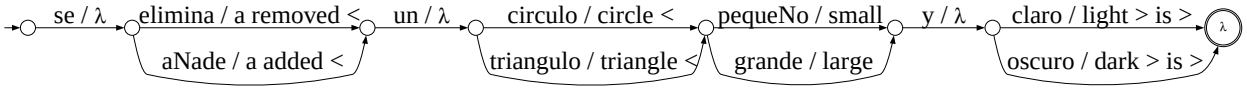
The Reordering Algorithm

Reordering is done by scanning the output sentence from left to right and creating a new reordered and bracketed sentence along the way.



Step	Word	Result (reordered and bracketed sentence)
1	a	a
2	large	a large
3	light	a large light
4	triangle	a triangle < large light >
5	is	a triangle < large light > is
6	removed	a removed < triangle < large light > is >

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 > "

Step	Word	Stack	Output
1	a	∅	a
2	removed <	removed	a
3	triangle <	removed, triangle	a
4	large	removed, triangle	a large
5	light	removed, triangle	a large light
6	>	removed	a large light triangle
7	is	removed	a large light triangle is
8	>	∅	a large light triangle is removed

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

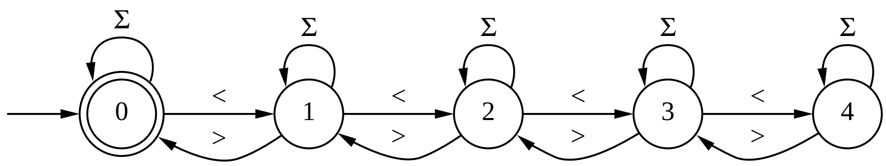
Possible problem:

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.



(Σ 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.

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

In brackets, model sizes (states/edges).

▷ ***Reordering can reduce the demand for training data by a factor of four***

Blank Page

Index

Statistical alignments and finite-state models

- Finite state transducer learning techniques seem to require large amounts of training data to produce acceptable results
- Some byproducts of statistical alignment model training can be useful to improve the learning capabilities of finite state methods:
  - ***Sentence-to-sentence word alignments***
  - ***Word-to-word mappings (statistical dictionaries)***

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

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

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

Statistical alignment models

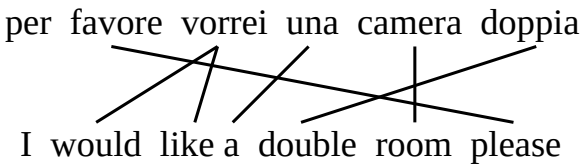
- **Alignments:** $a \subseteq \{1, ..., I\} \times \{1, ..., J\}, \quad I = |\, x \,|, \;\; J = |\, y \,|$
- Restriction: $a : \{1, ..., J\} \rightarrow \{0, ..., I\}$,

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

Example:

1	2	3	4	5	6
per	favore	vorrei	una	camera	doppia

I (0)	would (3)	like (3)	a (4)	double (6)	room (5)	please (2)
$a_1 = 0$	$a_2 = 3$	$a_3 = 3$	$a_4 = 4$	$a_5 = 6$	$a_6 = 5$	$a_7 = 2$



Review of OSTIA State-Merging Learning Procedures

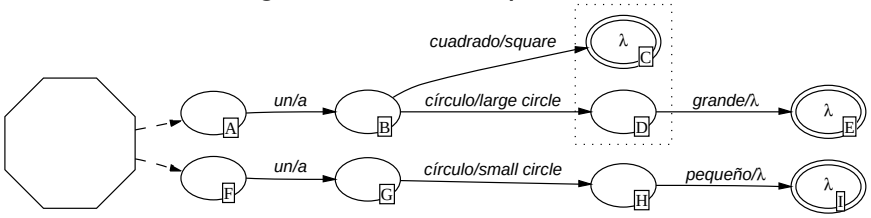
- Build an "*onward*" tree representation of the training data (a tree in which output strings are as close as possible to the root)
- The traversal of the tree goes in a level by level manner, typically by using a lexicographical order of state names.
- Two kinds of State Merging:
  - Merging based 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

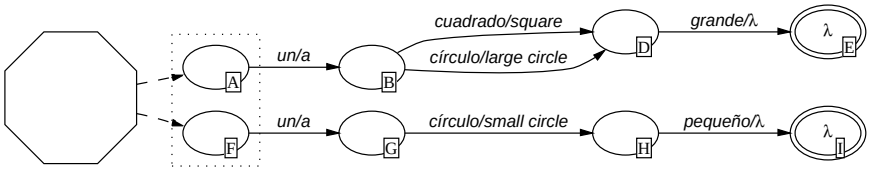
- OSTIA: only considers the output of the states: if both outputs are the same or at least one has no output, the join is possible *[Oncina, 91-93]*.
- OSTIA-DR: also takes into account two *Language Models* (LM), one for the Input (or Domain) and one for the Output (or Range): two states cannot be joined 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:

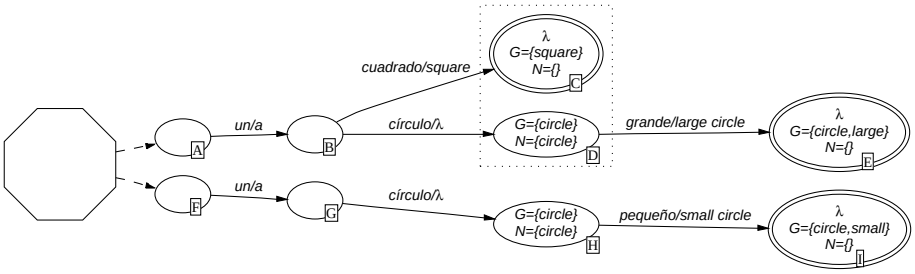


This entails a bad generalisation (un cuadrado grande, a square), and moreover now A and F could not be joined. This problem can be solved with a new extension to OSTIA called "***OMEGA***[1]"

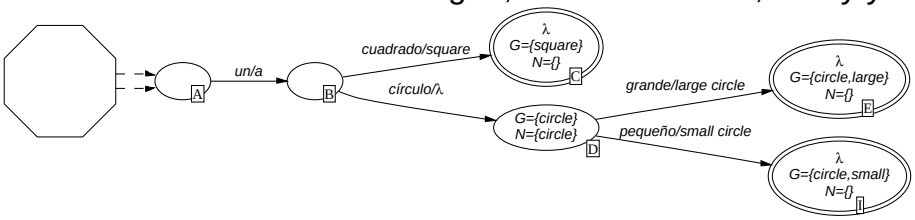
[1] For the Spanish "**O**STIA **M**odifi cado **E**mpleando **G**arant´ıas y **A**lineamientos" *[Vilar,98]*.

State Labeling with the help of a Dictionary and/or Alignment

Suposse that a known dicctionary or alignment hints that the translation of *grande*, *pequeño* and *círculo* 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*.
  - Alignements obtained using the ***MAR*** statistical model.
- **Search:** Error Correcting parsing.

Different Training Pairs	OSTIA-DR	OMEGA-DR
1,000	27,28	16,51
2,000	19,64	11,17
4,000	11,88	8,33
8,000	8,31	5,57
16,000	5,19	4,16

▷ ***Training data demands can be reduced by a factor of 2.***
▷ ***Results improve using Bilingual Categorization.***

Regular Grammars and finite state transducers: a morphism theorem

Theorem *[Berstel 1979]*:

$T \subseteq X^\star \times Y^\star$ is a rational translation if and only if there exist an alphabet Z, a regular language $L \subset Z^\star$ and two morphisms $h_X : Z^\star \to X^\star$ and $h_Y : Z^\star \to Y^\star$ 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 $X^\star \times Y^\star$ into a *single training string* $\mathbf{z}$ over an alphabet of *"extended symbols"* Z (composed of pairs of input/output symbols/strings)

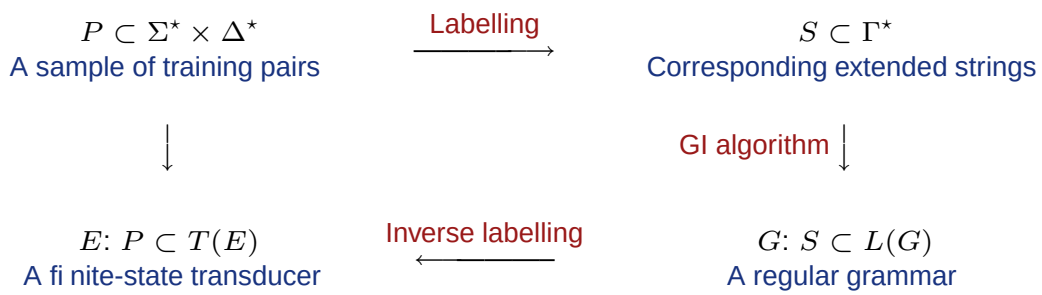
2. Use an adequate grammar learning technique (e.g., N-Grams) to obtain a finite state *"language model"* for these strings

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



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)

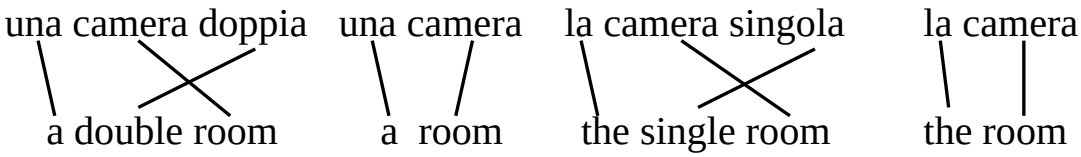
USING STATISTICAL ALIGNMENTS TO CONVERT TRAINING PAIRS INTO TRAINING STRINGS

Training pairs:

una camera doppia	→	a double room
una camera	→	a room
la camera singola	→	the single room
la camera	→	the room

Aligned sentences:

una camera doppia	una camera	la camera singola	la camera
a (1) double (3) room (2)	a (1) room (2)	the (1) single (3) room (2)	the (1) room (2)



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 $1 \leq i \leq |z|$

$$z_i = \begin{cases} x_i + y_j + y_{j+1} + \ldots + y_{j+l} & \text{if } \exists j : a(j) = i \text{ and } \neg\exists j' < j : a(j') > a(j) \\ & \text{and for } j'' : j \leq j'' \leq j + l, a(j'') \leq a(j) \\ x_i & \text{otherwise} \end{cases}$$

Aligned training pairs:

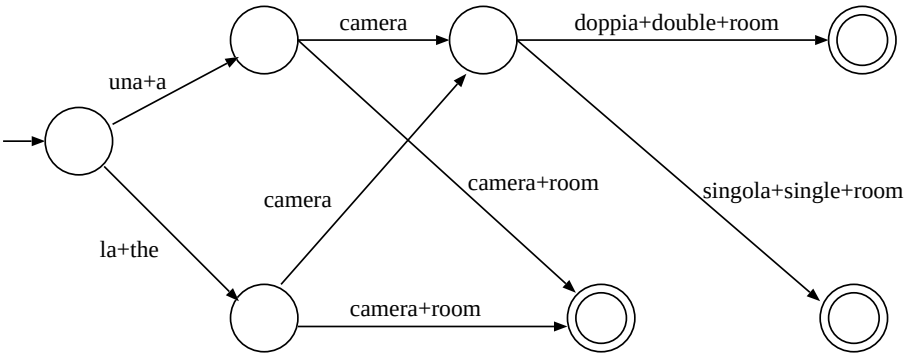
una camera doppia	a (1) double (3) room (2)	⇒	**una**+a **camera doppia**+double+room
una camera	a (1) room (2)	⇒	**una**+a **camera**+room
la camera singola	the (1) single (3) room (2)	⇒	**la**+the **camera singola**+single+room
la camera	the (1) room (2)	⇒	**la**+the **camera**+room

Training strings:

GIATI: Second step

FROM TRAINING STRINGS TO GRAMMARS: N-GRAMS

$$\Pr(z) \approx \prod_{i=1}^{|z|} \Pr(z_i | z_{i-n+1}, ..., z_{i-1})$$



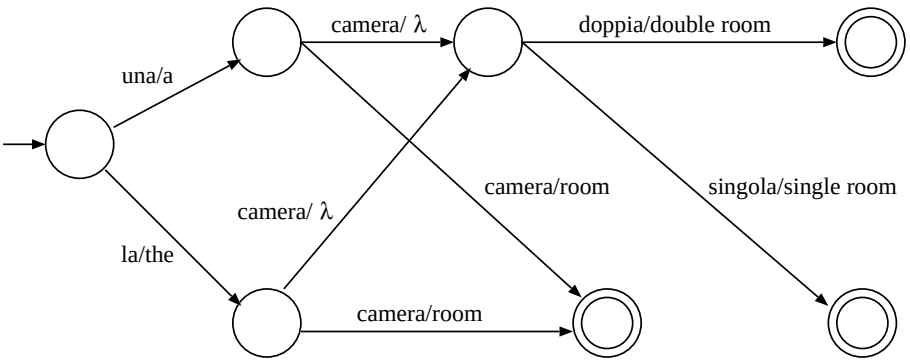
PROBLEM: Non-seen events in the training set.

COMMON SOLUTION: Smoothing.

GIATI: Third step

FROM GRAMMARS TO TRANSDUCERS: INVERSE LABELLING

GRAMMAR	TRANSDUCER
$(q, a + b_1 + b_2 + ... + b_k, q')$	$(q, a, b_1 b_2 ... b_k, q')$



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)

n	states	transitions	WER (%)	SER (%)
2	4,056	67,235	8.8	50.1
3	33,619	173,500	4.7	27.2
4	110,321	364,373	4.1	23.2
5	147,790	492,840	3.8	20.5
6	201,319	663,447	3.6	19.0
7	264,868	857,275	3.4	18.0
8	331,598	1,050,949	3.3	17.4
9	391,812	1,218,367	3.3	17.2
10	438,802	1,345,278	3.2	16.8
11	471,733	1,432,027	3.1	16.4
12	492,620	1,485,370	3.1	16.4

Comparative experiments: benchmark corpora

EUTRANS-I CORPUS [VIDAL 1997]

		Spanish	English
Train:	Sentences	10,000	
	Words	97,131	99,292
	Vocabulary	686	513
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)

		Italian	English
Train:	Sentences	3,038	
	Words	55,302	64,176
	Vocabulary	2,459	1,712
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]

Corpus	Method	Assited by	n-grams	**WER**
EuTrans-I	OSTIA	ECP	2	**8.3**
EuTrans-I	OMEGA	ECP, IBM2'	2	**6.6**
EuTrans-I	GIATI	BOS, IBM5	5	**6.6**
EuTrans-II	OMEGA	ECP, IBM2	2	**41.7**
EuTrans-II	OMEGA	ECP, IBM2, ABS	2	**36.5**
EuTrans-II	GIATI	BOS, IBM5	2	**28.1**
EuTrans-II	GIATI	BOS, IBM5, ABS	2	**24.9**

ECP = Error-Correcting Parsing
BOS = Back-Off Smoothing
ABS = Automatic Bilingual Segmentation
IBMk = IBM Model k statistical alignments
IBM2' = Symetrized IBM2

Summary of Stochastic Finite-State MT results

Translation Word Error Rate (TWER %)

Task	MLA	EuTrans-0	EuTrans-I	EuTrans-II	TT2-XRCE	AMETRA	TT2-UE
Languages	Sp-En	Sp-En	Sp-En	It-En	En-Sp	Sp-Ba	En-Sp
Vocabularies	30	689/514	689/514	2.5K/1.7K	26K/30K	719/1.3K	84K/97K
Training (words)	110K	4.5M	100K	50K	600K	90K	6M
Year	1993	1996	1998	1999	2004	2003	2004
OSTIA	3	$\approx$1	-	-	-	-	-
OSTIA-DR	1	$<$1	10	$>$80	-	-	-
OMEGA	-	$<$1	4	37	-	-	-
GIATI	-	3	7	25	32	40	56
Best result	-	-	4	25	28	36	47
Non FS system	-	-	AT	AT	PB	PB	PB

Languages: **En**glish, **Sp**anish, **It**alian, **Ba**sc

PB = Phrase-based alignment models

AT = Alignment Templates

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

Bibliography: FS transducers

- E.Vidal, P.García, E.Segarra: “Inductive Learning of Finite-State Transducers for Interpretation of Unidimensional Objects”. In“Structural Pattern Analysis,” pp.17-35. R.Mohr, T.Pavlidis, A.Sanfeliu, eds., World Scient. Pub., Series in Computer Science, 19, 1990.
- J.Oncina, P.García, E.Vidal: “Learning Subsequential Transducers for Pattern Recognition Interpretation Tasks”. IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol.PAMI-15, No.5, pp.448-458, 1993.
- E.Vidal: ”Language Learning, Understanding and Translation“. En “Progress and Prospects of Speech Research and Technology”, pp.131-140. H.Niemann, R.de Mori, G.Hanrieder (Eds.). Infi x, 1994.
- J.C.Amengual, E.Vidal: “Effi cient Error-Corecting Viterbi Parsing”. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.20, no. 10, 1998.
- A.Castellanos, E.Vidal, A.Varó, J.Oncina: “Language Understanding and Subsequential Transducer Learning”. Computer Speech and Language, No.12, pp.193-228. 1998.
- D.Picó and F.Casacuberta: “Some statistical-estimation methods for stochastic fi nite-state transducers”. Machine Learning, 44:121-142, 2001.
- E.Vidal, F.Thollard, C.de la Higuera, F.Casacuberta and R.C.Carrasco: “Probabilistic Finite-State Machines – Parts I and II” IEEE Trans on Pattern Analysis and Machine Intelligence (PAMI), 27(7):1013-1025, 2005.

Bibliography: State-merging FS learning

- J.Oncina, P.García, E.Vidal: “Learning Subsequential Transducers for Pattern Recognition Interpretation Tasks”. IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol.PAMI-15, No.5, pp.448-458, 1993.
- J.M.Vilar, E.Vidal, J.C.Amengual: “Learning Extended Finite-State Models for Language Translation”. Proc. of Extended Finite State Models Workshop (of ECAI’96), pp.92-96. Budapest, Agosto 1996.
- J.M.Vilar, V.M.Jiménez, J.C.Amengual, A.Castellanos, D.Llorens, E.Vidal: “Text and Speech Translation by means of Subsequential Transducers”. Natural Language Engineeering, Vol.2, No.4, pp.351-354, 1996.
- *E.Vidal*: “Finite-State Speech-to-Speech Translation”. Int. Conf. on Acoustics Speech and Signal Processing (ICASSP-97), proc., Vol.1, pp.111-114. Munich, 1997.
- A.Castellanos, E.Vidal, A.Varó, J.Oncina: “Language Understanding and Subsequential Transducer Learning”. Computer Speech and Language, No.12, pp.193-228. 1998.
- J.C.Amengual, J.M.Benedí, F.Casacuberta, A.Castaño, A.Castellanos, V.Jiménez, D.Llorens, A.Marzal, M.Pastor, F.Prat, E.Vidal, J.M.Vilar: “The EuTrans-I Speech Translation System”. Machine Translation. Vol.15, pp.75-103, 2000.

Bibliography: FS learning based on alignments

- J.M.Vilar: Improve the learning of subsequential transducers by using alignments and dictionaries. In “Grammatical Inference: Algorithms and Applications”, vol.1891 of *Lenture Notes in Artificial Intelligence*, pp.298–312. Springer-Verlag, 2000.

- F. Casacuberta: Inference of finite-state transducers by using regular grammars and morphisms. In “Grammatical Inference: Algorithms and Applications”, vol.1891 of *Lecture Notes in Artificial Intelligence*, pages 1–14. Springer-Verlag, 2000.

- F.Casacuberta and E.Vidal. Machine translation with inferred stochastic finite-state transducers. Computational Linguistics, 30(2):205-225, 2004.

- F.Casacuberta, E.Vidal, and D.Picó. Inference of finite-state transducers from regular languages. Pattern Recognition, In press, 2005.

Computational Intelligence and Learning Doctoral School
U. C. Louvain

Machine Translation:
Finite-State Models and Statistical Approaches

Speech-to-Speech Translation

Enrique Vidal

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

September 2007

E.Vidal – ITI-UPV-DSIC

Index

An utterance

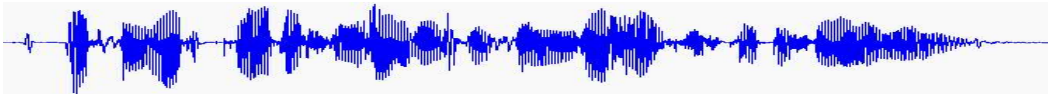
/por favor, quiero reservar una habitación doble hasta pasado mañana/

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 simpler text
- Speech categorization: From speech to class labels
- Speech understanding: From speech to “meaning” representation
- Dialog processing: From speech to “meaning” interactively
- Speech translation: From speech to speech in another language

Speech recognition, understanding and translation

INPUT:



SPEECH RECOGNITION OUTPUT:

por favor , quiero reservar una habitación doble hasta pasado mañana .

SPEECH UNDERSTANDING OUTPUT:

(ACTION=RESERVATION) (ROOM_TYPE=DOUBLE)
(DATE_OF_ENTRANCE=TODAY) (DATE_OF_LEAVING=TODAY+2)

SPEECH TRANSLATION OUTPUT:

I want to book a double room until the day after tomorrow, please.

Speech processing difficulties

- Noise and distortion
- No separation between adjancent words
- Words can be uttered in many different ways (even by the same speaker)
- Spoken sentences are often gramatically ill formed

Index

Statistical framework for speech recognition

Given an acoustic sequence v, search for an optimal sentence $\hat{x}$:

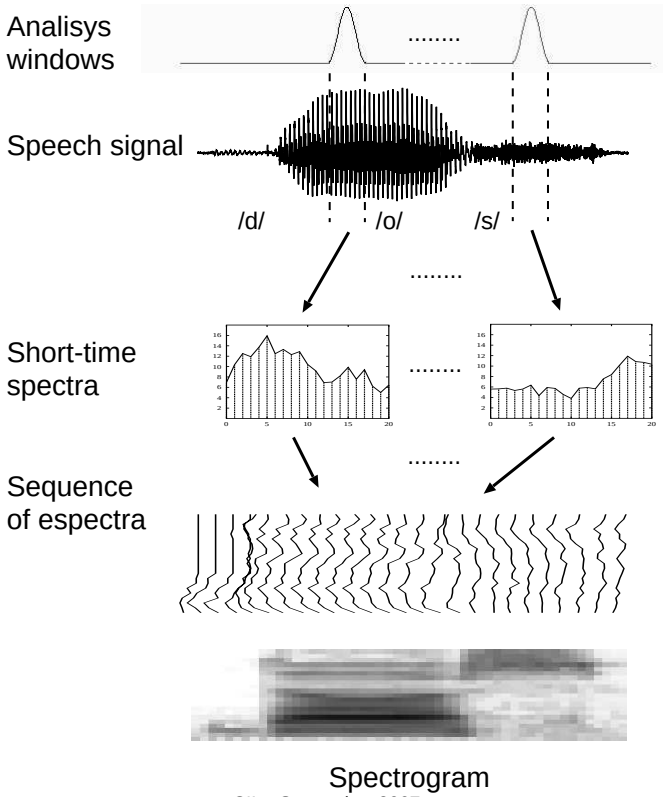
$$\hat{x} = \underset{s}{argmax} \Pr(x \mid v)$$

Using the Bayes' rule

$$\hat{x} = \underset{s}{argmax} \Pr(x) \cdot \Pr(v \mid x)$$

- SPEECH PREPROCESSING
  - Represent the speech signal as a sequence of acoustic vectors, v
- STATISTICAL MODELS FOR SPEECH RECOGNITION
  - $\Pr(v \mid x)$: **Acoustic models** (HIDDEN MARKOV MODELS)
  - $\Pr(x)$: **Language model** (N-GRAMS or STOCHASTIC GRAMMARS)

Speech preprocessing



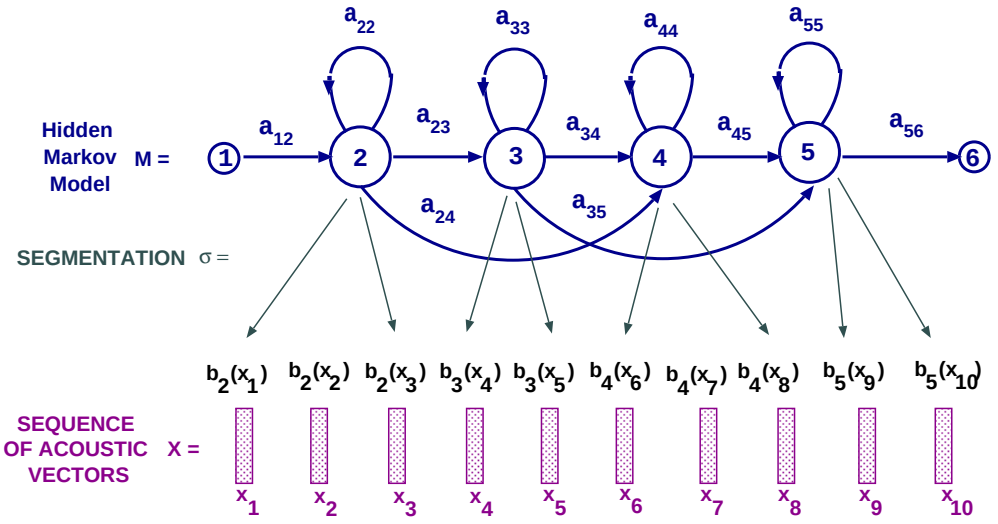
Spectrogram

Acoustic units

To model $\Pr(v \mid x)$, v (and x) must be decomposed into a sequence of **Acoustic-units**:

- **Words**:
  - \+ Include contextual effects (coarticulation)
  - – Too many units $\Rightarrow$ difficult training
- **Phonemes**:
  - \+ Small number of units $\Rightarrow$ easy training
  - – Coarticulation effects are not modeled
- **Compromise:**
  - $\star$ Adequate number of units,
  - $\star$ Include relevant coarticulation effects
  - $\star$ Proposals:
    sylabes, semi-sylabes, diphones, contextual phones, ...

Hidden Markov models (HMM)



$$Pr(x, \sigma \mid M) = a_{1,2} \cdot b_2(x_1) \cdot a_{2,2} \cdot b_2(x_2) \cdot a_{2,2} \cdot b_2(x_3) \cdot a_{2,3} \cdot b_3(x_4) \cdot a_{3,3} \cdot b_3(x_5) \cdot$$
$$a_{3,4} \cdot b_4(x_6) \cdot a_{4,4} \cdot b_4(x_7) \cdot a_{4,4} \cdot b_4(x_8) \cdot a_{4,5} \cdot b_5(x_9) \cdot a_{5,5} \cdot b_5(x_{10}) \cdot a_{5,6}$$

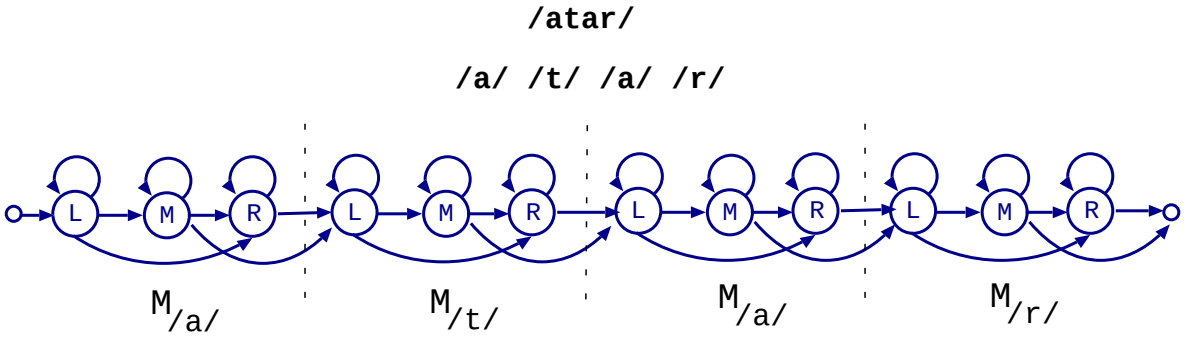
$$Pr(x \mid M) = \sum_{\sigma} Pr(x, \sigma \mid M)$$

Hidden Markov models (HMM)

- Components of a HMM $\mathcal{M} =< Q, E, a, \pi, b >$
  - Topology: Q: set of states. $E(= \Re^d)$: space of acoustic features.
  - Probabilistic distributions:
    - between states ($a : Q \times Q \rightarrow [0, 1]$),
    - initial state ($\pi : Q \rightarrow [0, 1]$)
    - emision (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, minimun discriminative information, discriminative.

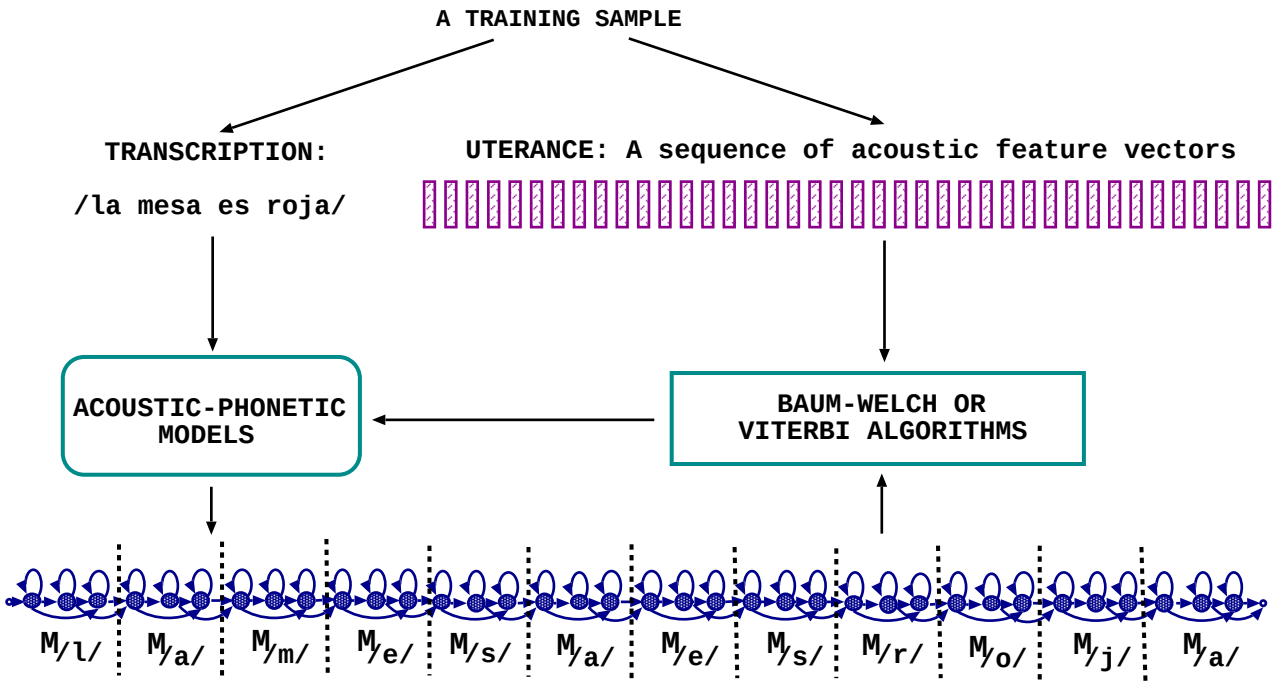
Modelling words in terms of phone acoustic models

Concatenation of phone units.



In a similar way, the utterance several connected words can be modelled as a concatenation of the models of these words

Training hidden Markov models



Language models

$$\Pr(x) = \prod_{i=1}^{I} \Pr(x_i \mid x_1^{i-1})$$

- Stochastic grammars $G = (N, \Sigma, R, S, p)$

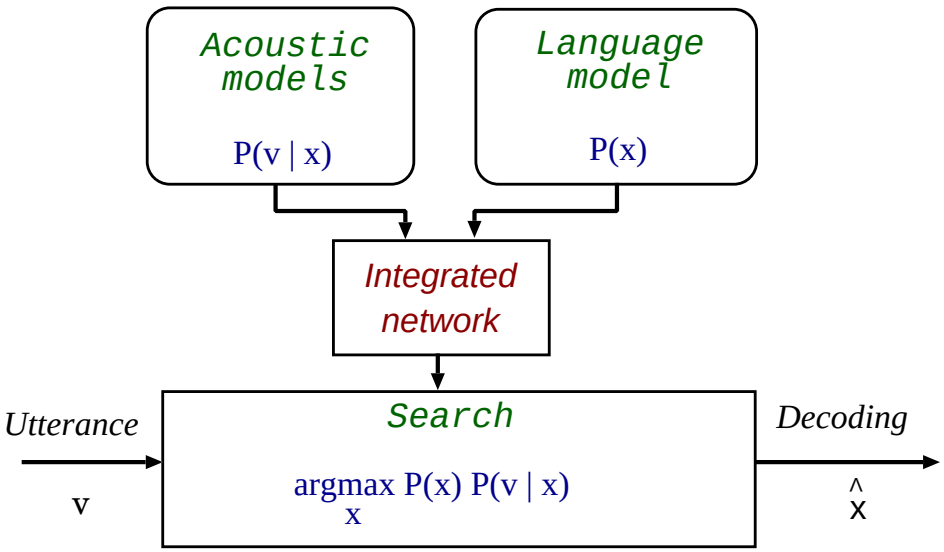
$$\Pr(x) \approx P_G(x) = \sum_{d \in d(x)} P_G(d)) \approx \max_{d \in d(x)} P_G(d))$$

(where $d(x)$ is the set of *derivations* of x in G)

- N-grams $\Pr(x) \approx \prod_{i=1}^{I} p_n(x_i \mid x_{i-n+1}^{i-1})$

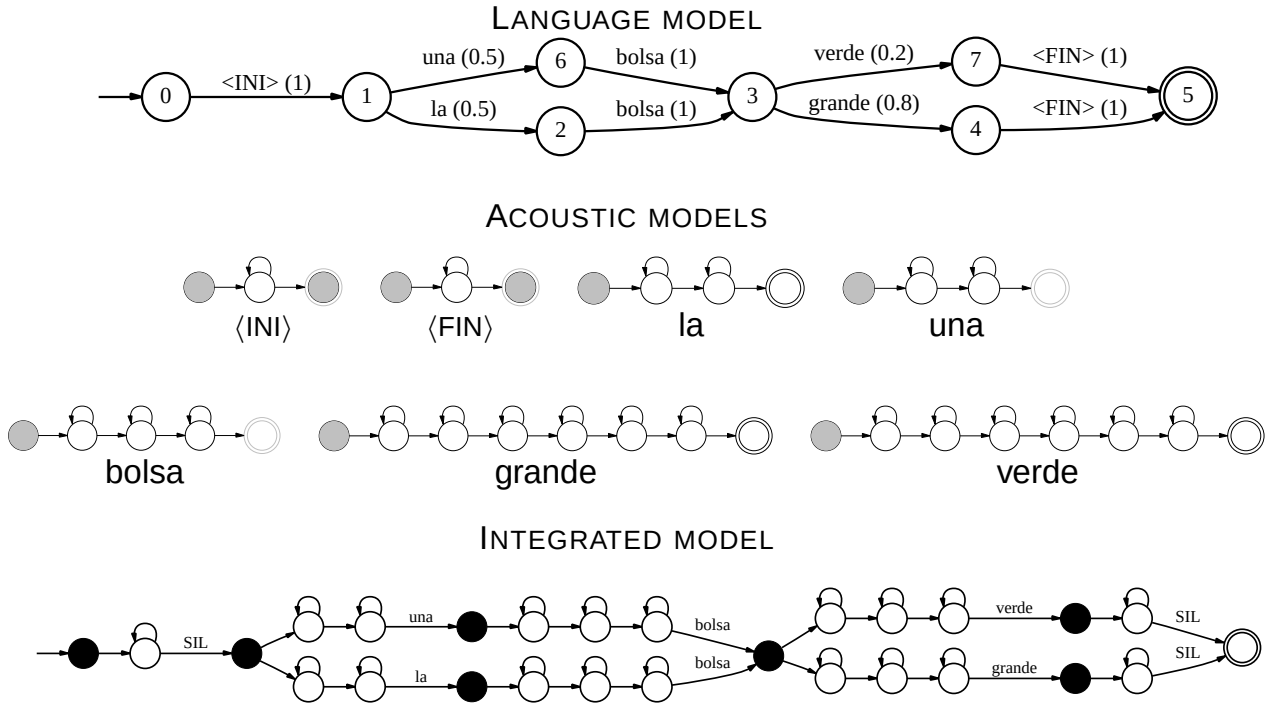
- Learning:
  - Grammatical inference techniques
  - Maximum likelihood, maximum entropy
  - Smoothing
  - Extensions: categories, cache, triggers, etc.

Integrated architecture for speech recognition

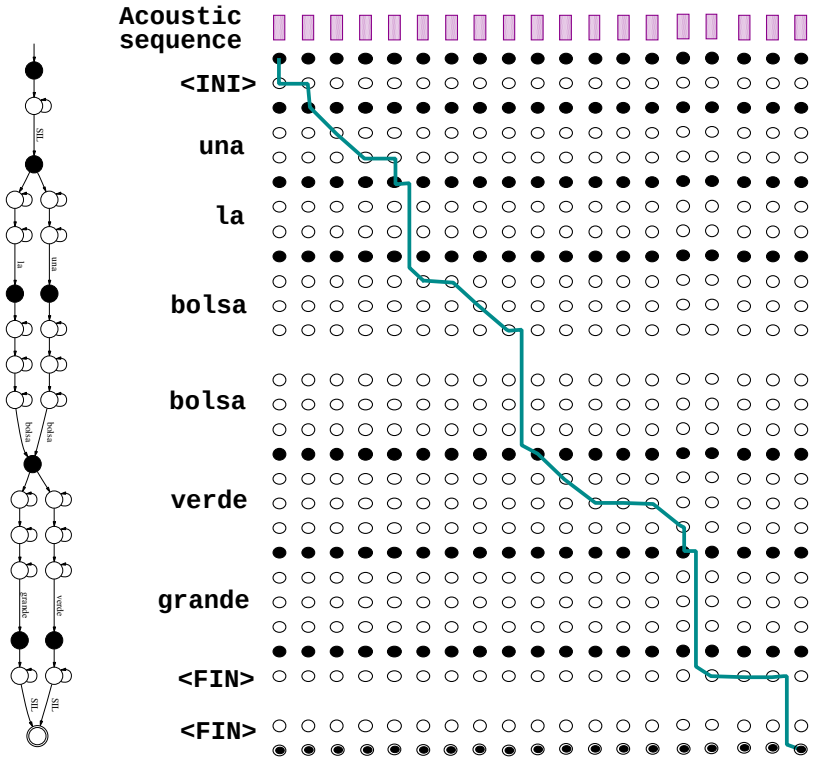


Search engine:
THE VITERBI ALGORITHM (+ beam search + ...)

Integrated architecture for speech decoding



An example of speech decoding



$$\underset{x}{argmax}\,(Pr(x)\cdot Pr(v|x))$$

$\approx$ "una bolsa verde"

Index

Limited Domain Speech-to-Speech Translation

Issues

\+ Tasks with *small or medium-sized vocabularies* and *restricted semantic scope*.

– *Non-grammatical, spontaneous sentences* expected.

– *Robust systems needed*, accepting *text* and/or *speech* input.

– *No manual "pre/post-editing"* is possible.

– Only *low development costs* can be afforded.

General statistical framework for speech translation

Given an acoustic sequence v, search for the target sentence $\hat{y}$:

$$\hat{y} = \underset{y}{argmax} \Pr(y \mid v)$$

The translation can be viewed as a two-step process:

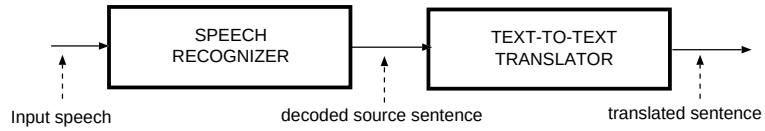
$$v \longrightarrow x \longrightarrow y$$

where x is any possible decoding of v, and y is the translation of x.

$$\underset{y}{argmax} \Pr(y \mid v) = \underset{y}{argmax} \sum_{x} \Pr(y, x \mid v) \approx \underset{y}{argmax} \underset{x}{max} \Pr(y, x \mid v)$$

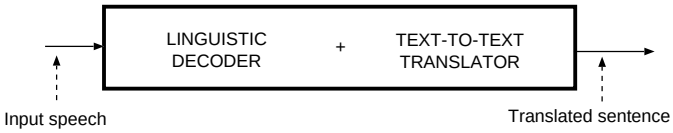
Speech input translation

CONVENTIONAL APPROACH: **SEQUENTIAL**



- *Problem*: The speech recognizer is error prone.
- *Advantage*: Low computational costs.

ALTERNATIVE: **INTEGRATED**



- *Advantage*: Conceptually simpler & potentially more robust.
- *Problem*: Computational costs might be prohibitive in general (but are *low with FS transducers*).

Speech translation using SFSTs

$$\hat{y} \;=\; \underset{y}{argmax}\, \underset{x}{max} \Pr(y, x \mid v) \;=\; \underset{y}{argmax}\, \underset{x}{max} \left(\Pr(x, y) \cdot \Pr(v \mid x)\right)$$

- $\Pr(v|x)$: Acoustic models
  - HIDDEN MARKOV MODELS
- $\Pr(x, y)$: Translation models
  - STOCHASTIC FINITE-STATE TRANSDUCERS

INTEGRATED ARCHITECTURE TO SPEECH TRANSLATION.

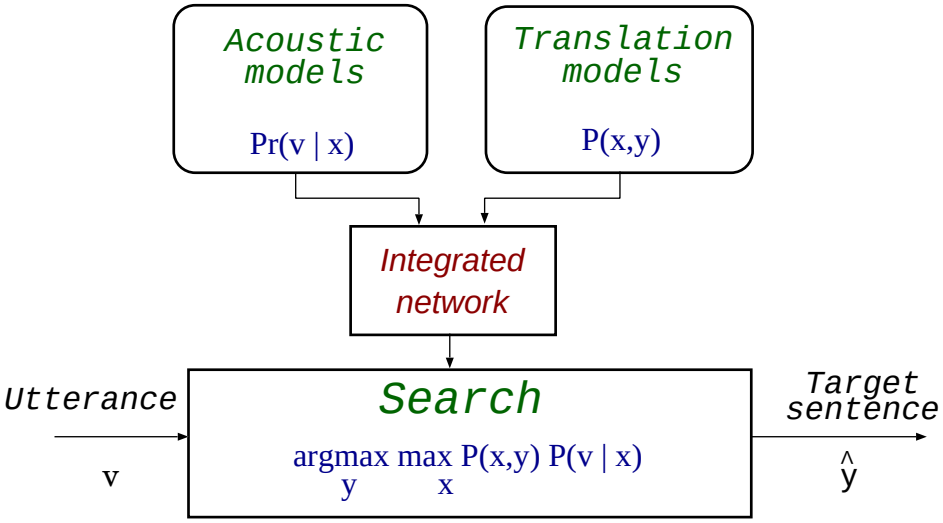
Integration

If $\Pr(x, y)$ and $\Pr(v \mid x)$ are modeled by a *FS transducer* and *acoustic HMMs*, respectively, then the search problem $argmax_y\, max_x \left(\Pr(x, y) \cdot \Pr(v \mid x)\right)$ can be optimally solved by Dynamic Programming.

- Integration simply consists in an expansion of the FS fransducer arcs with the acustic HMMs (also FS models).
- The integrated network fully embodies the input-language *acoustic constraints*, the input and output-language *syntactic* constraints and the *input/output word mappings*.
- The conventional (beam-search accelerated) Viterbi algorithm yields the hypothesis for a best sequence of arcs of the transducer.

Viterbi search directly yields best hypothesis for both output (translated) and input (recognized) sentences.

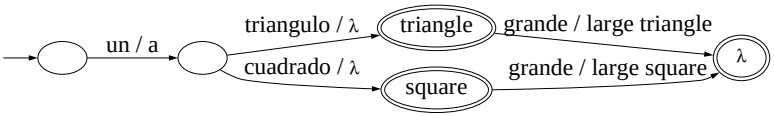
Integrated architecture for speech translation



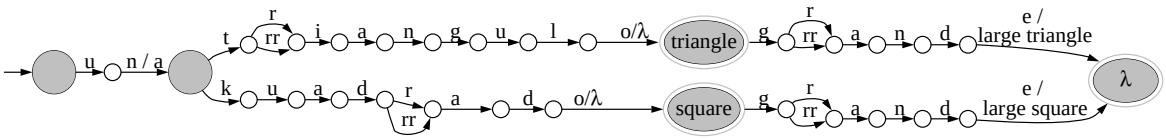
Search engine:
THE VITERBI ALGORITHM (+ beam search + ...)

Illustration of the integration process

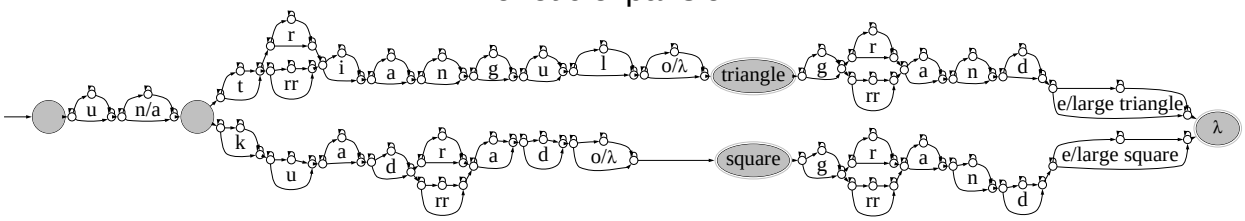
Original Finite State Transducer:



Lexical expansion:

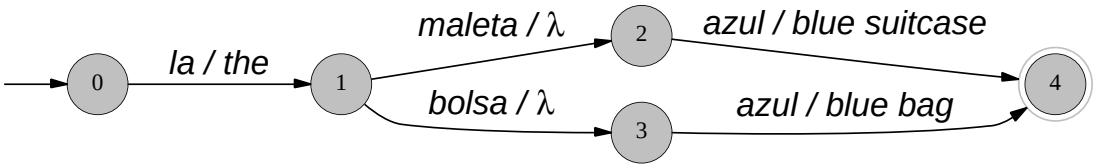


Phonetic expansion:

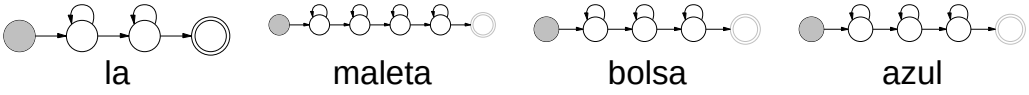


Integration process: another example

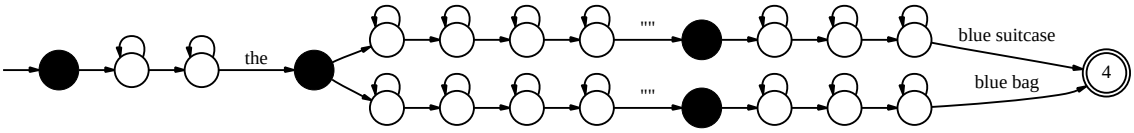
ORIGINAL FINITE-STATE TRANSDUCER



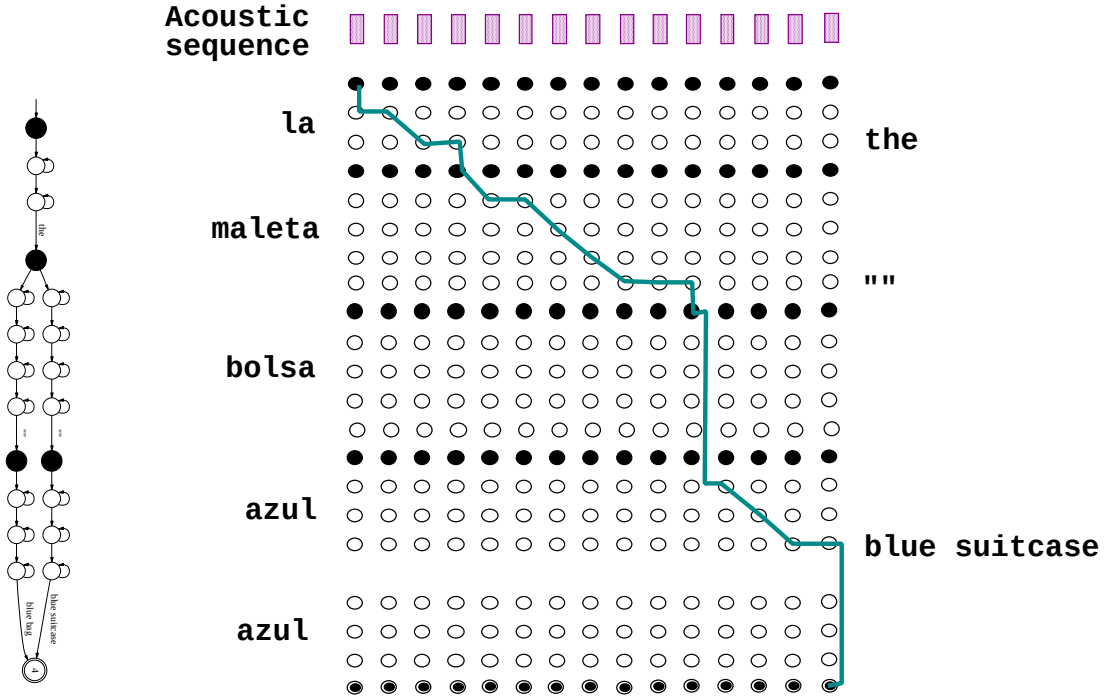
ACOUSTIC MODELS



PHONETIC EXPANSION



An example of integrated speech translation with SFST



$argmax_{y,x}\, Pr(v \mid x) \cdot Pr(y, x)$ ≈"the blue suitcase / la maleta azul"

Another framework for statistical speech translation

$$\hat{y} \;=\; \underset{y}{argmax}\, \underset{x}{max}\, \Pr(y, x \mid v) \;=\; \underset{y}{argmax}\, \underset{x}{max}\, (\Pr(y \mid x) \cdot \Pr(x) \cdot \Pr(v \mid x))$$

- $\Pr(v|x)$: Acoustic models
  - HIDDEN MARKOV MODELS
- $\Pr(x)$: Source language models
  - N-GRAMS
- $\Pr(y \mid x)$: Translation models
  - STOCHASTIC FINITE-STATE TRANSDUCERS
  - STATISTICAL ALIGNMENT MODELS + STOCHASTIC DICTIONARIES

SERIAL ARCHITECTURE TO SPEECH TRANSLATION.

Serial architecture for speech translation

$$\begin{aligned} \hat{y} \;&=\; \underset{y}{argmax}\, \underset{x}{max}\, \{Pr(y \mid x) \cdot Pr(x) \cdot Pr(v \mid x)\} \\ &\approx\; \underset{y}{argmax}\, \Pr(y \mid \hat{x})\,, \quad \hat{x} \;=\; \underset{x}{argmax}\, \Pr(x) \cdot \Pr(v \mid x) \end{aligned}$$

1. *Word decoding of* v: $\hat{x} = \underset{x}{argmax}\, \{Pr(x) \cdot Pr(v|x)\}$

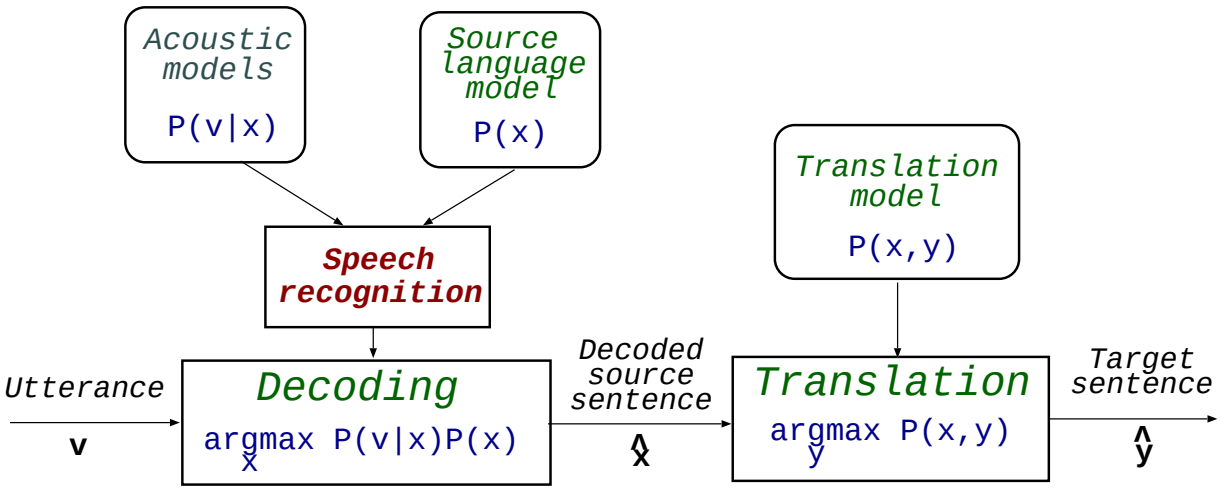
   $Pr(x)$: source language model; $Pr(v|x)$: acoustic models.

2. *Translation of* $\hat{x}$:

   $$\hat{y} = \underset{y}{argmax}\, Pr(y \mid \hat{x}) = \underset{y}{argmax}\, Pr(y, \hat{x}) = \underset{y}{argmax}\, Pr(\hat{x} \mid y) \cdot Pr(y)$$

   $Pr(y, \hat{x})$ or $Pr(\hat{x} \mid y)$: translation model; $Pr(y)$: target language model.

Serial architecture for speech translation



Search engine for decoding and text translation:
THE VITERBI ALGORITHM (+ beam search + ...)

Yet another architecture: Iterative search

$$\hat{y} = \underset{y}{argmax}\, \Pr(y \mid v) \approx \underset{y}{argmax}\, \underset{s}{max} \{\Pr(y) \cdot \Pr(x \mid y) \cdot \Pr(v \mid x)\}$$

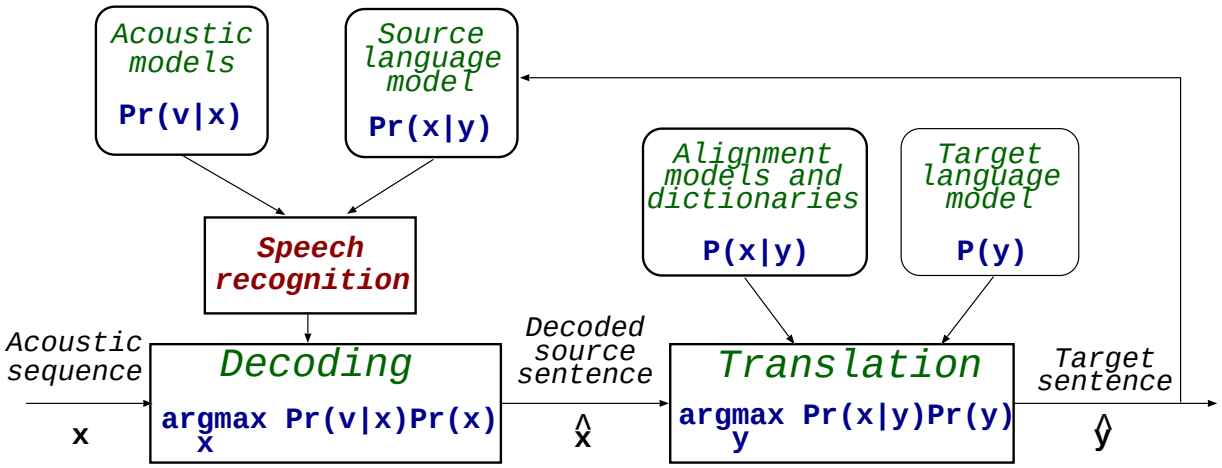
a) INITIALIZATION

1. *Decoding* v: $\hat{x} \approx \underset{x}{argmax} \{\Pr(x) \cdot \Pr(v|x)\}$
2. *Translating* $\hat{x}$: $\hat{y} \approx \underset{y}{argmax} \{\Pr(\hat{x}|y) \cdot \Pr(y)\}$

b) GENERAL ITERATION

1. *Decoding* v *using* $\hat{y}$: $\hat{x} \approx \underset{x}{argmax} \{\Pr(x \mid \hat{y}) \cdot \Pr(v|x)\}$
2. *Translating* $\hat{x}$: $\hat{y} \approx \underset{y}{argmax} \{\Pr(\hat{x}|y) \cdot \Pr(y)\}$

Iterative search architecture



Index

MTA Speech-Input Results: Impact of Integration and Input/Output Syntactic Constraints

[Jiménez, Castellanos and Vidal, ICASSP-95]

- OSTIA–DR transducers trained on $50,000$ MLA Spanish–English sentences, with *4–Gram* Input (Spanish) and/or Output (English) LMs.
- Edges of the learned transducers expanded with simple acoustic-phonetic *discrete HMMs* representing the corresponding input words.
- 400 test utterances (4 speakers, 100 sentences not used in training)

Language Model Usage	Text Transl. WER	Speech Recogn. WER	Speech Transl. WER
NO LMS (only the basic SST)	0.0%	98.0%	**96.5%**
DECOUPLED (Input LM front-end)	—	4.8%	**15.2%**
INTEGRATED: (NPUT LM ONLY)	0.0%	3.5%	**3.7%**
INTEGRATED: (INPUT/OUTPUT LMS)	0.0%	2.6%	**2.8%**

*Translation **robustness** improves dramatically with **integrated models**, and more so as more **syntactic constraints** are included.*

MTA Speech-Input Results: Impact of using Partial Alignment and Word Reordering

[Vilar, Vidal and Llorens, SPECOM-96]

OSTIA–DR learning with *reordered-bracketed* training pairs, using input language 4-gram constraints and output constraints enforcing balanced bracketing: Model size (Edges) and translation Word Error Rate (WER in %) for increasing size of the training–set. *Test set*: 400 unseen, speaker-independent utterances.

	Direct		Reordered	
Train.Pairs	Edges	WER	Edges	WER
1,000	2,023	**45.5**	1,338	**17.5**
2,000	3,353	**38.7**	979	**7.3**
4,000	4,051	**28.2**	440	**3.2**
8,000	719	**4.8**	344	**3.0**
16,000	363	**3.2**	183	**3.3**

*Only 4,000 training pairs are required to achieve similar results as with the direct approach with at least **four times** more training data.*

Traveler Task Speech-Input Experiments: Impact of Integration

[Amengual et al., Machine Translation Journal 2000]

- Comparison between two schemes (text training set: full EUTRANS-0 corpus):
  - *Integrated*: OSTIA-DR SSTs, learned with *Categories* and I/O *3-Gram constraints*.
  - *Serial*: *first*, speech recognition using the input language 3-Gram LM and *then*, translation through a (good) SST learned with no domain/range constraints.
- *Acoustic-Phonetic modeling*: 25 context-independent continuous density HMMs (2,462 gausians) trained from 21K words by 20 speakers.
- *Test–Set*: 336 unseen utterances, 3K words (84 sentences, uttered by 4 speakers).

Approach	Recognition Word Error Rates	Translation Word Error Rates
Serial	2.15 %	3.54 %
Integrated	1.98 %	1.83 %

- *High degree of recognition/translation* ***robustness*** *achieved thanks to the* tight integration *of acoustic, syntactic and translation models.*
- *Takes advantage of the* lower perplexity of the output language *to actually* ***improve the overall translation accuracy!***

Comparative experiments: benchmark speech corpora

EUTRANS-I Corpus

Data		Spanish	English
Training text	Sentence pairs	10,000	
	Different sentence pairs	6,813	
	Running words	132,198	134,922
	Vocabulary	686	513
	Bigram Test-Set Perplexity	8.6	6.3
Training speech	Running words	11,000	—
Test	Speech utterances	336	—
	Running words	3,000	—

Comparative experiments: benchmark speech corpora

EuTrans-II Corpus

Data		Italian	English
Training text	Sentence pairs (all different)	3,038	
	Running words	61,232	72,446
	Vocabulary	2,459	1,701
	Bigram Test-Set Perplexity	31	25
Training speech	Running words	52,511	—
Test	Speech utterances	278	—
	Running words	5,381	—

Comparative results on the EuTrans-I benchmark speech corpus

Input	Models	Architecture	Source LM	WER	TWER
Mic	ALTEMP	serial	trigrams	4.1	6.9
	GIATI	serial	trigrams	4.1	7.8
		integrated	GIATI	4.4	7.9
	OMEGA	serial	trigrams	4.1	12.7
		integrated	OMEGA	13.6	12.5
Tel	ALTEMP	serial	trigrams	11.6	13.3
	GIATI	serial	trigrams	11.6	14.1
		integrated	GIATI	10.5	12.6
	OMEGA	serial	trigrams	11.6	17.7
		integrated	OMEGA	18.3	17.9

Mic = microphone, Tel = telephone, WER = ASR WER, TWER = Translation WER

Comparative results on the EuTrans-II benchmark speech corpus

Models	Architecture	Source LM	WER	TWER	SSER
ALTEMP	serial	trigrams	22.1	29.5	38.7
GIATI	serial	trigrams	22.1	29.5	40.3
	integrated	GIATI	32.0	35.5	-
OMEGA	serial	trigrams	22.1	41.6	-
	integrated	OMEGA	52.5	46.2	-

Only telephone, WER = ASR WER, TWER = Translation WER,
SSER = Subjective Sentence Error rate

EuTrans demos

On-line demos
http://prhltdemos.iti.es/demo/

Index

Bibliography

1. Jelinek, *Statistical Methods for Speech Recognition*. The MIT Press, 1998.
2. Rabiner. *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
3. Ney, *Speech translation: Coupling of recognition and translation*, Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP99), 1999.
4. Casacuberta and de la Higuera. *Linguistic decoding is a difficult computational problem*. Pattern Recognition Letters, 20:813–821, 1999.
5. Amengual, Benedí, Casacuberta, Castaño, Castellanos, Jiménez, Llorens, Marzal, Pastor, Prat, Vidal, and Vilar, *The EuTrans-I Speech Translation System*. Machine Translation, 15:75–103, 2000.
6. Ney, Niessen, Och, Sawaf, Tillmann, and Vogel, *Algorithms for statistical translation of spoken language*. IEEE Transactions on Speech and Audio Processing, 8(1):24–36, 2000.
7. Casacuberta, Ney, Och, Vidal, Vilar, Barrachina, García-Varea, Llorens, Martínez, Molau, Nevado, Pastor, Picó, Sanchis, and Tillmann, *Some approaches to statistical and finite-state speech-to-speech translation*. Computer Speech and Language, 18:25–47, 2004.
8. Casacuberta, Vidal, Sanchis, and Vilar. *Pattern recognition approaches for speech-to-speech translation*. Cybernetic and Systems: an International Journal, 35(1):3–17, 2004.
9. I.García-Varea, A.Sanchis, and F.Casacuberta, *A decoding algorithm for speech input statist. translation*. TSD-2004. Vol.3206 of LNCS, pp.305-314, Springer-Verlag, 2004.

Computational Intelligence and Learning Doctoral School
U. C. Louvain

Machine Translation:
Finite-State Models and Statistical Approaches

Computer Assisted Translation

Enrique Vidal

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

September 2007

E.Vidal – ITI-UPV-DSIC

Index

Introduction to Computer Assited 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 $\hat{y}_s$ *accpted* by the user ($\mathbf{a}$), followed by text obtained by means of additional user keystrokes ($\mathbf{k}$) 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 throughly 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:

ITER-0	(y_p)	( )
ITER-1	$(\hat{y}_s)$	(*Haga clic para cerrar el diálogo de impresión*)
	$(\mathbf{a})$	(Haga clic)
	$(\mathbf{k})$	(**en**)
	(y_p)	(Haga clic en)
ITER-2	$(\hat{y}_s)$	(*ACEPTAR para cerrar el diálogo de impresión*)
	$(\mathbf{a})$	(*ACEPTAR para cerrar el*)
	$(\mathbf{k})$	(**cuadro**)
	(y_p)	(Haga clic en ACEPTAR para cerrar el cuadro)
FINAL	$(\hat{y}_s)$	(*de diálogo de impresión*)
	$(\mathbf{a})$	(*de diálogo de impresión*)
	$(\mathbf{k})$	(#)
	$(y_p \equiv y)$	(Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión)

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.

Index

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 sufixes:

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

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

$$\begin{aligned} \hat{y}_s &= \underset{y_s}{argmax} \Pr(x, y_p, y_s) \\ &= \underset{y_s}{argmax} \Pr(x, y_p y_s) \quad (1) \\ &= \underset{y_s}{argmax} \Pr(x \mid y_p y_s) \cdot \Pr(y_p y_s) \quad (2) \end{aligned}$$

- (1) Stochastic Finite State Transducers
- (2): Statistical Alignment and Language models
- Text-input MT is a paricular case, where $y_p = \lambda$
- Main difference of CAT vs. MT: search over the set of suffixes

CAT Interactive Search

High speed is needed because typically a new system hypothesis must be produced in real time after each user keystroke

WORD-GRAPH 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

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 Prefi x (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 adapten 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 p

H: Feed direction for standard paper sizes 1-9

T: Feed direction for standard paper sizes 1-9

More examples of CAT human-machine (keyboard) interaction

S: Edición de la lista de impresoras

H: Editing printers

P: Editing t

H: Editing the printers

P: Editing the printer l

H: Editing the printer list

T: Editing the printer list

Index

Benchmark Xerox printer manuals corpus

Data		English	Spanish	English	German	English	French
Train	Sent. pairs	56K		53K		49K	
	Run. words	572K	657K	543K	583K	507K	441K
	Vocabulary	26K	30K	25K	27K	25K	37K
	Voc-Simp	8K	12K	7K	19K	8K	10K
Test	Sentences	1 125		984		996	
	Run. words	7.6K	9.4K	9.6K	10.0K	10.8K	9.8K
	Out of Voc.	341	362	219	552	252	255
	Run. chars.	46K	58K	55K	63K	61K	71K
	Perplexity	107	60	93	169	193	135

Voc-Simp corresponds to vocabulary sizes after tokenizarion, case normalization, etc.

Benchmark EU bulletin corpus

Data		English	Spanish	English	German	English	French
Train	Sent. pairs	214K		223K		215K	
	Run. words	5.9M	6.6M	6.5M	6.1M	6.0M	6.6M
	Vocabulary	84K	97K	87K	152K	85K	91K
Test	Sentences	800		800		800	
	Run. words	20K	25K	22K	21K	22K	24K
	Out of Voc.	108	140	107	227	113	119
	Perplexity	96	72	95	153	97	71

CAT results with the Xerox corpus

DATA:	GIATI 3-gram (1-best)			GIATI 3-gram (5-best)		
XRCE2	KSR	CER	TWER	KSR	CER	TWER
En-Es	17.6	30.3	43.1	15.6	25.0	37.8
Es-En	21.5	35.5	51.4	18.9	28.1	45.2
En-Fr	37.1	54.3	73.8	34.3	48.5	69.6
Fr-En	39.4	55.3	71.9	36.7	49.5	67.7
En-De	38.8	62.8	81.3	35.4	56.7	77.2
De-En	36.4	61.5	78.5	32.9	55.1	73.3

CAT results with the EU corpus

DATA:	GIATI 5-gram (1-best)			GIATI 5-gram (5-best)		
EU	KSR	CER	TWER	KSR	CER	TWER
En-Es	27.5	37.6	55.8	24.6	34.8	51.7
Es-En	25.4	38.0	52.5	22.7	35.1	48.0
En-Fr	26.2	36.0	53.9	23.5	33.4	50.1
Fr-En	23.1	36.1	49.2	20.6	32.8	44.4
En-De	29.4	41.2	65.5	26.8	38.1	60.3
De-En	31.0	44.4	66.6	28.0	41.4	61.2

Index

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.

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:

$$\hat{y} = \underset{y}{argmax} \Pr(y \mid x, v)$$

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

$$\hat{y} = \underset{y}{argmax} \Pr(v \mid y) \cdot \Pr(x \mid y) \cdot \Pr(y) = \underset{y}{argmax} \Pr(v \mid y) \cdot \Pr(y \mid x)$$

- $\Pr(v \mid y) \approx$ (TARGET LANGUAGE) ACOUSTIC MODELS
- $\Pr(x \mid y) \approx$ TRANSLATION MODEL
- $\Pr(y) \approx$ TARGET LANGUAGE MODEL
- $\Pr(y \mid x) \approx$ TARGET "LM" CONDITIONED BY SOURCE TEXT

Similar to plain speech decoding, where: $\hat{y} = argmax_y \Pr(v \mid y) \cdot \Pr(y)$

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:

$$\hat{y}_s = \underset{y_s}{argmax} \Pr(y_s \mid x, y_p) \tag{3}$$

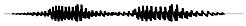
The user is now allowed to *utter some words*, v, generally aimed at amending parts of $\hat{y}_s$ and the system has then to obtain a most probable decoding of v:

$$\hat{d} = \underset{d}{argmax} \Pr(d \mid x, y_p, \hat{y}_s, v) \tag{4}$$

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:

ITER-0	(y_p)	( )
ITER-1	$(\hat{y}_s)$	(*Haga clic para cerrar el diálogo de impresión*)
	(v)	
	$(\hat{d})$	(**Haga clic a**)
	(**k**)	(**en ACEPTAR**)
	(y_p)	(Haga clic en ACEPTAR)
ITER-2	$(\hat{y}_s)$	(*para cerrar el diálogo de impresión*)
	(v)	
	$(\hat{d})$	(**cerrar el cuadro)**
	(**k**)	( )
	(y_p)	(Haga clic en ACEPTAR para cerrar el cuadro)
FINAL	$(\hat{y}_s)$	(*de diálogo de impresión*)
	(**k**)	(**#**)
	$(y_p \equiv y)$	(**Haga clic** en ACEPTAR para **cerrar el cuadro** de diálogo de impresión)

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):

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

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

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

- $\Pr(v \mid d)$ $\approx$ (TARGET LANGUAGE) ACOUSTIC MODELS
- $\Pr(d \mid x, y_p, \hat{y}_s) \approx$ 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:

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

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:

$$\hat{d} = \underset{d}{argmax} \Pr(d \mid x, y_p) \cdot \Pr(v \mid d)$$

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:

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

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:

$$\hat{d} = \underset{d}{argmax} \Pr(d \mid \hat{y}_s) \cdot \Pr(v \mid d)$$

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 form 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 constrined by the known prefix (source text ignored)

	DEC	DEC-PREF	CAT-PREF	CAT-SEL
Word Error Rate (%)	18.6	16.1	10.6	1.6
Sentence Error rate (%)	50.2	44.4	30.0	3.6

Using knowledge about the source sentence is more important than using only user-validated prefixes

Bibliography

- G. Foster, P. Langlais, G. Lapalme. User-Friendly Text Prediction for Translators. Conference on EMNLP. 2002.
- F.Casacuberta and E.Vidal. Machine translation with inferred stochastic finite-state transducers. Computational Linguistics, 30(2):205-225, 2004.
- J. Civera, J. Vilar, E. Cubel, A. Lagarda, F. Casacuberta, E. Vidal, D. Picó, and J. González, A syntactic pattern recognition approach to computer assisted translation. Advances in Statistical, Structural and Syntactical Pattern Recognition – S+SSPR 2004 IAPR workshop. A. Fred, T. Caelli, A. Campilho, R. P. Duin, and D. de Ridder, Eds. LNCS, Springer-Verlag, Lisbon, 2004.
- E.Vidal, F.Casacuberta, L.Rodr´ıguez, J.Civera and C.Mart´ınez Computer-Assisted Translation Using Speech Recognition. To be published, 2005.