Universitat Politècnica de València
Master en Inteligencia Artificial, Renoconocimiento de Formas e Imagen Digital
Curs 2006-2007

SYSTEMS AND TOOLS FOR MACHINE TRANSLATION

GIZA++: Training of statistical translation models

Francisco Casacuberta Enrique Vidal

fcn@iti.upv.es evidal@iti.upv.es

June 4, 2007

IARFID-UPV

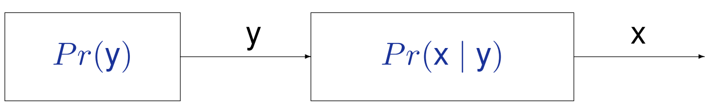
Index

Index

Statistical alignment models

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

A “distorted (noisy) channel model”



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

Alignments

- **Alignments:** (Brown et al. 90) $J = |\mathrm{x}|$ y $I = |\mathrm{y}|$

$$\mathrm{a} : \{1, ..., J\} \rightarrow \{0, ..., I\}$$

  $\mathrm{a}_j = 0 \Rightarrow j$ in x is not aligned with any position in y.
- Set of possible alignments: $\mathcal{A}(\mathrm{x}, \mathrm{y}) = \{\mathrm{a} : \{1, ..., J\} \rightarrow \{0, ..., I\}\}$
- The probability of translation y to x through an alignment a is $\Pr(\mathrm{x}, \mathrm{a} \mid \mathrm{y})$

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

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

Model 1

$$\Pr(\mathrm{x}, \mathrm{a} \mid J, \mathrm{y}) = \prod_{j=1}^{J} \Pr(\mathrm{a}_j \mid \mathrm{a}_1^{j-1}, J, \mathrm{y}) \cdot \Pr(\mathrm{x}_j \mid \mathrm{x}_1^{j-1}, \mathrm{a}, J, \mathrm{y})$$

- $\Pr(\mathrm{a}_j \mid \mathrm{a}_1^{j-1}, J, \mathrm{y}) \approx \frac{1}{(I+1)^J}$
- $\Pr(\mathrm{x}_j \mid \mathrm{x}_1^{j-1}, \mathrm{a}, J, \mathrm{y}) \approx l(\mathrm{x}_j \mid \mathrm{y}_{\mathrm{a}_j})$ **statistical lexicon**

$$P_{M1}(\mathrm{x} \mid \mathrm{y}) = \frac{n(J|I)}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} l(\mathrm{x}_j \mid \mathrm{y}_i)$$

Model 2

$$\Pr(\mathsf{x}, \mathsf{a} \mid J, \mathsf{y}) = \prod_{j=1}^{J} \Pr(\mathsf{a}_j \mid \mathsf{a}_1^{j-1}, J, \mathsf{y}) \cdot \Pr(\mathsf{x}_j \mid \mathsf{x}_1^{j-1}, \mathsf{a}, J, \mathsf{y})$$

- $\Pr(\mathsf{a}_j \mid \mathsf{a}_1^{j-1}, J, \mathsf{y}) \approx a(\mathsf{a}_j \mid j, J, I)$ **statistical alignments**
- $\Pr(\mathsf{x}_j \mid \mathsf{x}_1^{j-1}, \mathsf{a}, J, \mathsf{y}) \approx l(\mathsf{x}_j \mid \mathsf{y}_{\mathsf{a}_j})$ **statistical lexicon**

$$P_{M2}(\mathsf{x} \mid \mathsf{y}) = n(J|I) \cdot \prod_{j=1}^{J} \sum_{i=0}^{I} a(i \mid j, J, I) \cdot l(\mathsf{x}_j \mid \mathsf{y}_i)$$

Homogeneous HMM alignment

$$\Pr(\mathsf{x}, \mathsf{a} \mid J, \mathsf{y}) = \prod_{j=1}^{J} \Pr(\mathsf{a}_j \mid \mathsf{a}_1^{j-1}, J, \mathsf{y}) \cdot \Pr(\mathsf{x}_j \mid \mathsf{x}_1^{j-1}, \mathsf{a}, J, \mathsf{y})$$

- $\Pr(\mathsf{a}_j \mid \mathsf{a}_1^{j-1}, J, \mathsf{y}) \approx h(\mathsf{a}_j \mid \mathsf{a}_{j-1}, J, I)$ **statistical alignments**
- $\Pr(\mathsf{x}_j \mid \mathsf{x}_1^{j-1}, \mathsf{a}, J, \mathsf{y}) \approx l(\mathsf{x}_j \mid \mathsf{y}_{\mathsf{a}_j})$ **statistical lexicon**

$$P_{HMM}(\mathsf{x} \mid \mathsf{y}) = n(J \mid I) \cdot \sum_{\mathsf{a}} \prod_{j=1}^{J} h(\mathsf{a}_j \mid \mathsf{a}_{j-1}, J, I) \cdot l(\mathsf{x}_j \mid \mathsf{y}_{\mathsf{a}_j})$$

Optimal alignment with Model 2

Search for the "best" alignment from $\mathcal{A}(\mathsf{x}, \mathsf{y})$

$$\widehat{\Pr}(\mathsf{x} \mid \mathsf{y}) \quad \approx \quad \Pr(J \mid \mathsf{y}) \cdot \max_{\mathsf{a} \in \mathcal{A}(\mathsf{y},\mathsf{x})} \Pr(\mathsf{x}, \mathsf{a} \mid J, \mathsf{y})$$

Using Model 2,

$$\widehat{P}_{M2}(\mathsf{x} \mid \mathsf{y}) \quad = \quad n(J \mid I) \cdot \prod_{j=1}^{J} \max_{0 \le i \le I} [a(i \mid j, J, I) \cdot l(\mathsf{x}_j \mid \mathsf{y}_i)]$$

Viterbi algorithm $(\mathsf{x}, \mathsf{y}, l, a)$

For $j := 1$ **until** J $\quad A[j] := \operatorname{argmax}_{0 \le i \le I} a(i \mid j, J, I) \cdot l(\mathsf{x}_j \mid \mathsf{y}_i)$ **End-for**

Return: A

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

Model 3

$$\Pr(\mathsf{x} \mid \mathsf{y}) = \sum_{\mathsf{a}} \Pr(\mathsf{x}, \mathsf{a} \mid \mathsf{y}) = \sum_{\mathsf{a}} \sum_{(\phi,\tau,\pi) \in \mathcal{F}(\mathsf{x},\mathsf{a})} \Pr(\phi, \tau, \pi \mid \mathsf{y})$$

The probability for a tablet τ and a permutation π is:

$$\Pr(\phi, \tau, \pi \mid \mathsf{y}) = \Pr(\phi \mid \mathsf{y}) \cdot \Pr(\tau \mid \phi, \mathsf{y}) \cdot \Pr(\pi \mid \tau, \phi, \mathsf{y})$$

- $f(\phi_i \mid \mathsf{y}_i)$ *fertility probability*
- $l(x \mid \mathsf{y}_i)$ *lexicon probability*
- $d(j \mid i, J, I)$ *distortion probability*

$$P_{M3}(\mathsf{x} \mid \mathsf{y}) =$$

$$\sum_{a_1=0}^{I} \cdots \sum_{a_J=0}^{I} \binom{J-\phi_0}{\phi_0} p_0^{J-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{I} \phi_i! \cdot f(\phi_i \mid \mathsf{y}_i) \prod_{j=1}^{J} l(\mathsf{x}_j \mid \mathsf{y}_{\mathsf{a}_j}) \cdot d(j \mid \mathsf{a}_j, J, I)$$

Model 4

The center of a target word y_i, $c(i) = \frac{\sum_k \pi_{i,k}}{\phi_i}$

- $f(\phi_i \mid \mathsf{y}_i)$ *fertility probability*
- $l(x \mid \mathsf{y}_i)$ *lexicon probability*
- $d_{=1}(j - c(i-1) \mid \mathcal{C}_{\mathcal{Y}}(\mathsf{y}_{i-1}), \mathcal{C}_{\mathcal{X}}(\mathsf{x}_j))$

  distortion probability for the first position in a tablet
- $d_{>1}(j - \pi_{i,k-1} \mid \mathcal{C}_{\mathcal{X}}(\mathsf{x}_j))$

  distortion probability for the rest of positions in a tablet

Model 5

For a target word y_i:

- For a target word y_i:, number of vacant positions up to and including position j just before $\tau_{i,k}$ is placed, $v(j, \tau_1^{i-1}, \tau_{i,1}^{k-1}) \equiv v_j$.

- $f(\phi_i \mid \mathsf{y}_i)$ *fertility probability*
- $l(x \mid \mathsf{y}_i)$ *lexicon probability*
- $d_{=1}(v_j \mid \mathcal{C}_{\mathcal{X}}(\mathsf{x}_j), v_{c(i-1)}, v_J - \phi_i + 1) \cdot (1 - \delta(v_j, v_{j-1}))$

  distortion probability for the first position in a tablet
- $d_{>1}(v_j - v_{\pi_{i,k-1}} | \mathcal{C}_{\mathcal{X}}(\mathsf{x}_j), v_J - v_{\pi_{i,k-1}} - \phi_i + k) \cdot (1 - \delta(v_j, v_{j-1}))$

  distortion probability for the rest of positions in a tablet

The training process

- Every model has a specific set of free parameters.
- For example for IBM Model 4: $\theta = \left\{ \{l(x|y)\}, \{p_{=1}(\Delta_j)\}, \{p_{>1}(\Delta_j)\}, \{p(\phi|x)\}, p_1 \right\}$
- To train the model parameters θ: A maximum likelihood criterium, using a parallel training corpus consisting of S sentence pairs $\{(\mathrm{x}^{(n)}, \mathrm{y}^{(n)}) : n = 1, \ldots, N\}$:

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \prod_{n=1}^{N} \sum_{\mathrm{a}} p_\theta(\mathrm{x}^{(n)}, \mathrm{a}|\mathrm{y}^{(n)}) \quad .$$

- The training is carried out using the Expectation-Maximization (EM) algorithm.

The training process

- Maximum likelihood by EM estimation.
- The counts in the reestimation are multiplied by $Pr_M(\mathrm{x}, \mathrm{a} \mid \mathrm{y})$ and are added for all possible alignment.
- No efficient method is computing these estimated counts.
- The estimated counts are approximate by:
  - Computing the (approximate) most probable alignment (Model 2)
  - Apply modifications: moves and swaps
  - Sum the estimated counts for all alignments whose probability is larger than the probability of the probable alignment times a given constant.
  - More details: P. F. Brown et al. *The mathematics of statistical machine translation: parameter estimation.* Computational Linguistics, vol. 19 (2), 263–310, 1993.

Index

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++).

  `http://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 the program GIZA

  Old version: `http://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
...

Here is an 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.

Here's a sample of 3 sentences:

```
...
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

- This is optional. The dictionary file is of the 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":

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)
```


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.4.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 $P(i|j,l,m)$ 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 position of i and j are switched:

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

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)