Statistical Machine Translation

Marcello Federico
FBK-irst Trento, Italy
Galileo Galilei PhD School – University of Pisa

Pisa, 7-19 May 2008

Part VI: Phrase-Based Systems

- Log-linear Model Framework
- Estimation of Weights
- Log-Linear Phrase-based Models
- Extraction of phrase-pairs
- Open source toolkit Moses

Discriminative Approach to SMT

Log-Linear Model for word-alignment MT approach:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \sum_{\mathbf{a}} \Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \approx \arg\max_{\mathbf{e}} \max_{\mathbf{a}} \Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \tag{1}$$

$\Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f})$ is determined through real valued feature functions $h_k(\mathbf{e}, \mathbf{f}, \mathbf{a})$, $k = 1 \ldots M$, and takes the parametric form:

$$p_{\boldsymbol{\lambda}}(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) = \frac{\exp\{\sum_k \lambda_k h_k(\mathbf{e}, \mathbf{f}, \mathbf{a})\}}{\sum_{\mathbf{e},a} \exp\{\sum_{k'} \lambda_{k'} h_{k'}(\mathbf{e}, \mathbf{f}, \mathbf{a})\}} \tag{2}$$

Special case: feature functions give standard IBM model:

$$\begin{array}{lcll} h_1(\mathbf{e}, \mathbf{f}, \mathbf{a}) & = & \log \Pr(\mathbf{e}) & \text{(language model)} \\ h_2(\mathbf{e}, \mathbf{f}, \mathbf{a}) & = & \log \Pr(\mathbf{a} \mid \mathbf{e}) & \text{(distortion model)} \\ h_3(\mathbf{e}, \mathbf{f}, \mathbf{a}) & = & \log \Pr(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) & \text{(lexicon model)} \end{array}$$

Search Criterion and Properties

The **search criterion of MT** can be rewritten as:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \max_{\mathbf{a}} \sum_k \lambda_k h_k(\mathbf{e}, \mathbf{f}, \mathbf{a})\} \tag{3}$$

The Log-Linear framework gives the following **advantages**:

- directly models the posterior probability (discriminative model)
- does not rely on probability factorizations with independence assumptions
- its mathematically sound framework permits to add any kind of feature
- includes any IBM-model as special case, e.g. see previous slide with λ set to 1
- MLE or minimum error training can be applied to estimate free parameters (λ)
- Features used during search are decomposable wrt to target string, e.g. M4:
  $h(\mathbf{f}, \mathbf{a}, \mathbf{e}) = \sum_i^l h_i(\tau_i, \pi_i, \tilde{e}_i)$
  in this way, scores can be computed incrementally.

Training of Log-Linear Models

Instead of applying MLE, training can directly address performance optimization:

$$\boldsymbol{\lambda}_* = \arg\min_{\boldsymbol{\lambda}} E_D(\boldsymbol{\lambda}) \tag{4}$$

where $E_D(\boldsymbol{\lambda})$:

- measures translation errors over a development set D, e.g. BLEU or NIST
- can be very irregular, i.e. has many local minima

We apply multi-variate minimization. E.g. the simplex algorithm, which:

- empirically evaluates $E_D(\boldsymbol{\lambda})$ several times until convergence
- requires running the SMT search algorithm for each evaluation

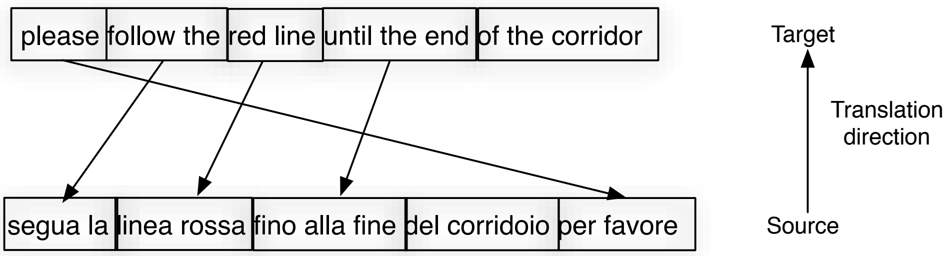
Log-linear phrase-based SMT

- Translation hypotheses are ranked by a **log-linear combination of statistics**:

$$\mathbf{e}^* = \operatorname*{argmax}_{\mathbf{e}} \ \max_{\mathbf{b}} \sum_k \lambda_k h_k(\mathbf{e}, \mathbf{f}, \mathbf{b})$$

- Phrases are finite sequence of words: n-grams with no linguistic meaning
- **Hidden variable** $\mathbf{b}$ represents segmentation and re-ordering:
  - segmentation maps the source text into a sequence of phrases
  - source phrases are translated into target phrases
  - alignment defines the order of translation
- **Feature functions** include:
  - Lexicon Model: table of phrase-pair translations
  - Distortion Model: word movement of consecutive phrases
  - Language Model: fluency of target words in target phrases
  - Length Model: to bias longer target strings

From Word-based to Phrase-based Alignment



Target-to-source alignment shows that:

- Source text is segmented into phrases
- Source phrases are translated in different order
  – e.g. `per favore` is translated first!
- Each source phrase is translated with a target phrase

Phrase-based Feature Functions

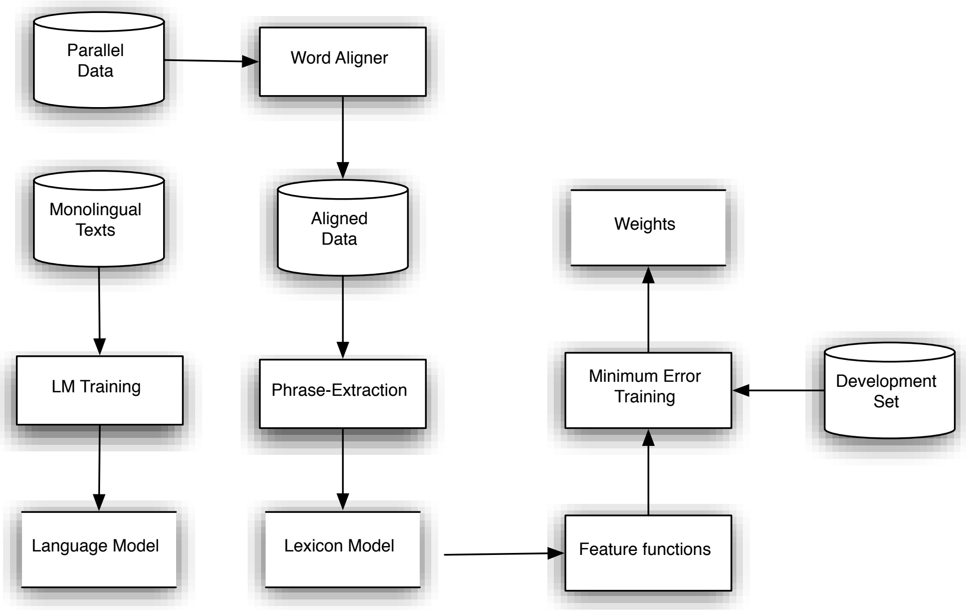
- **Feature decomposition** is necessary to perform DP search:

$$h_k(\mathbf{e}, \mathbf{f}, \mathbf{b}) = \sum_{i}^{l} h_k(\tilde{f}_{b_i}, b_i, \tilde{e}_i; b_{i-1}, \tilde{e}_{i-1})$$

  $\tilde{e}_i =$ i-th target phrase, $b_i = (b_i^f, b_i^l) =$ first/last positions covered by $\tilde{e}_i$
  $\tilde{f}_{b_i} =$ source phrase translated by $\tilde{e}_i$

- Lexicon Model: dir/inv relative freq, dir/inv IBM M1 probs
  $h_1 = \log \frac{N(\tilde{f},\tilde{e})}{N(\tilde{f})}$ $\quad h_2 = \log \frac{N(\tilde{f},\tilde{e})}{N(\tilde{e})}$
  $h_3 = \log \Pr_{M1}(\tilde{f} \mid \tilde{e})$ $\quad h_4 = \Pr_{M1}(\tilde{e} \mid \tilde{f})$
- Distortion model: $h_5 = \log(\exp(-dist(b_i, b_{i-1})))$
- Language Model: $h_6 = \log p(\tilde{e}_i \mid \tilde{e}_{i-1})$ **more than one LM can be used!**
- Length Model: $h_7 = len(\tilde{e}_i)$

Training of Log-Linear Models



Log-Linear Phrase Modeling: Search

Partial translation hypotheses are build bottom-to-top over the source positions:

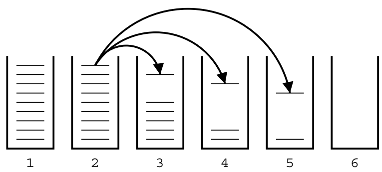
- **Alignments must cover all positions of $\mathbf{f}$ exactly once**
  (cf. TSP constraint: each city has to be visited exactly once)

- **Alternatives are explored and scored along different directions**:
  – possible segmentations of input into phrases
  – possible translations of source phrases
  – possible re-ordering of source phrases
- **Search algorithms**: A^* and DP beam search
- **Approximations**: limited word-reordering and translation options

Log-Linear Phrase Modeling: Moses Decoder

- **Phrase-based decoding steps:**
  - **cover** some not yet covered consecutive words (span)
  - **retrieve** phrase-translations for the span
  - **compute** translation, distortion and target language models

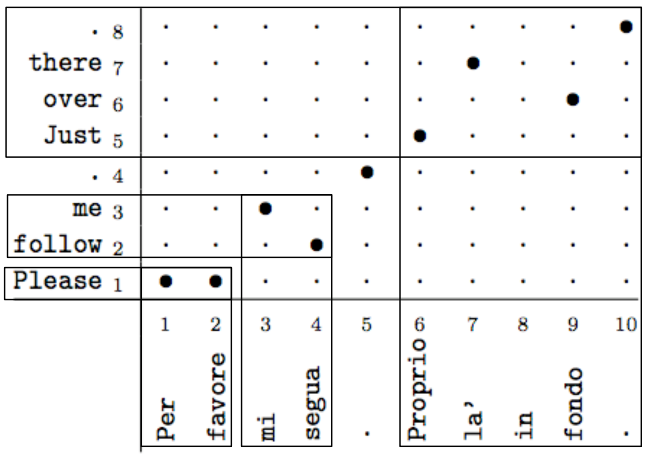
- **Multi-stack decoder**:
  - theories stored according to the coverage size
  - synchronous on the coverage size



- **DP recombination** of similar partial translations
- **Beam search**:
  - deletion of less promising partial translations:
  - histogram and threshold pruning
- **Distortion limit**: reduction of possible alignments
- **Lexicon pruning**: limit the amount of translation options per span

Alignments and Phrases

- Word alignment can be used to extract **phrase-pairs**
  - per favore - please
  - proprio la' in fondo . - just over there .
  - mi segua - follow me



Important: alignments of words in the phrase-pair must all be the rectangle, with the exception of null word alignments.

mi segua - follow is not a valid phrase-pair!

Phrase-pairs Extraction

Given a pair $(\mathbf{f}, \mathbf{e})$ and:

- direct alignment: $\mathbf{a} = \{(j, a_j) : \; j = 1, \ldots, m \; \wedge \; a_j \in \{0, \ldots, l\}\}$
- inverted alignment: $\mathbf{b} = \{(b_i, i) : \; i = 1, \ldots, l \; \wedge \; b_i \in \{0, \ldots, m\}\}$

We can compute symmetric alignments:

- union: $\mathbf{c} = \{(j, i) : 1 \leq j \leq m, \;\; 1 \leq i \leq l \;$ s.t. $a_j = i \; \cup \; b_i = j\}$
- intersection: $\mathbf{d} = \{(j, i) : 1 \leq j \leq m, \;\; 1 \leq i \leq l \;$ s.t. $a_j = i \; \cap \; b_i = j\}$
- grow-diagonal: enrich $\mathbf{d}$ with selected links from $\mathbf{a}$ or $\mathbf{b}$ (Moses)

Properties:

- $\mathbf{a}$ and $\mathbf{b}$ are maps betwen two sets of positions
- $\mathbf{c}$ is a many-to-many partial alignment between $\mathbf{f}$ to $\mathbf{e}$ (we exclude null words)
- $\mathbf{d}$ is a 1-1 partial alignment between $\mathbf{f}$ to $\mathbf{e}$
- we will exploit $\mathbf{c}$ and $\mathbf{d}$ to extract phrase-pairs between $\mathbf{f}$ and $\mathbf{e}$

Phrase-pairs Extraction

Given a pair $(\mathbf{f}, \mathbf{e})$ and any alignment $\mathbf{c}$, let $J = [j_1, j_2]$ and $I = [i_1, i_2]$ denote two closed intervals within the positions of $\mathbf{f}$ and $\mathbf{e}$, respectively.

- Let $\mathbf{c}_e[J]$ denote the set of target positions linked to source positions J by $\mathbf{c}$:

$$c_e[J] = \{i : \exists j \in J \text{ s.t. } (j, i) \in \mathbf{c}\} \tag{5}$$

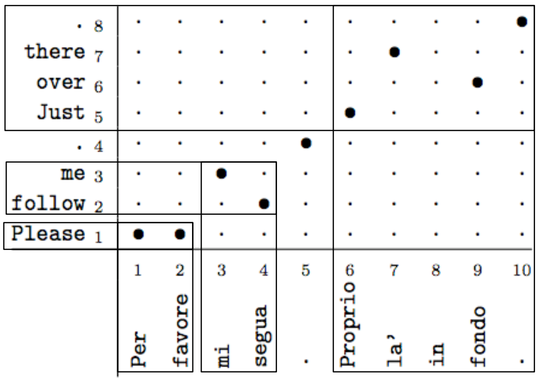
- Let $\mathbf{c}_f[I]$ denote the set of source positions linked to target positions I by $\mathbf{c}$:

$$c_f[I] = \{j : \exists i \in I \text{ s.t. } (j, i) \in \mathbf{c}\} \tag{6}$$

- We say that I and J form a phrase-pair $((\tilde{f}, \tilde{e}) = (f_{j_1}^{j_2}, e_{i_1}^{i_2}))$ under $\mathbf{c}$ iff $\mathbf{c}$ links all positions in J into I and all positions in I into J, i.e.:

$$\emptyset \subset \mathbf{c}_e[J] \subseteq I \;\wedge\; \emptyset \subset \mathbf{c}_f[I] \subseteq J \tag{7}$$

Phrase-pairs Extraction




```
per favore -- please
mi segua -- follow me
.  -- .
per favore mi segua -- please follow
me
per favore mi segua .  -- please
follow me .
proprio la' in fondo -- Just over
there
...
```

You can extract phrase-pairs similarly from any type of alignment.

[Exercise 2. Find all phrase-pairs]

Phrase-pairs Extraction

Given a training sample provided with best direct and indirect alignments

$$\{\mathbf{f}^s, \mathbf{e}^s, \mathbf{a}^s, \mathbf{b}^s) : s = 1, \ldots, S\}$$

through alignments $\mathbf{c}^s$ or $\mathbf{d}^s$ we can derive a large collection of phrase-pairs:

$$\mathcal{P} = \{(\tilde{f}^p, \tilde{e}^p) : p = 1, \ldots, P\}$$

where $\tilde{f}$ and $\tilde{e}$ indicate, respectively, word sequences of $\mathcal{E}$ and $\mathcal{F}$.

- $\mathcal{P}$ is extended with single word phrases from $\mathbf{a}$
- Store phrase-pairs in a translation phrase-table and compute probabilities:
  – from relative frequency counts in both directions
  – from word-based translation probabilities by applying Model 1

Phrase-based Language Model

- **Phrases**: build from words by interleaving the symbol #
  Example: `I#would#like`.
- **Phrase Probability**: expand phrases into words and apply word-based LM:

$$\Pr(\tilde{e}_1 = e_{1,1}\# \ldots \# e_{1,k_1} \mid \tilde{e}_3, \tilde{e}_2) = \Pr(e_{1,1} \mid \tilde{e}_3, \tilde{e}_2) \prod_{t=2}^{k_1} \Pr(e_{1,i} \mid \tilde{e}_3, \tilde{e}_2, \tilde{e}_{1,1}, \ldots, \tilde{e}_{1,i-1})$$

- **Conditional part**:
  – phrases are expanded into word sequences
  – n-gram history is cut to the depth of the word-based LM

Moses Toolkit for Statistical MT

- Developed during **JHU Summer Workshop 2006**
  – U. Edinburgh, ITC-irst Trento, RWTH Aachen, U. Maryland, MIT, Charles University Prague
  – open source under Lesser GPL
  – available for Linux, Windows and Mac OS
  – www.statmt.org/moses
- **Main features**:
  – translation of both text and CN inputs
  – exploitation of more Language Models
  – lexicalized distortion model
  – incremental pre-fetching of translation options from disk
  – handling of huge LMs (up to Giga words)
  – on-demand and on-disk access to LMs and TMs
  – factored translation model (surface forms, lemma, POS, word classes, ...)