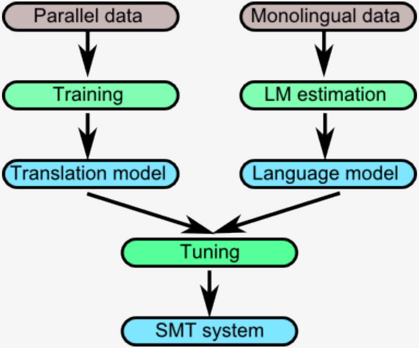
Statistical Machine Translation

-discriminative training-

Artem Sokolov
Computerlinguistik
Universität Heidelberg
Sommersemester 2015

material from P. Koehn, F. Yvon



Tuning stage

- closest to output $\Rightarrow$ high impact
- sweet spot for ML researchers
  (can be agnostic of the above engineering details)
- bulk of MT research happens here

- previously all models we used were generative
- would require modeling $p(\mathbf{f})$ or $p(\mathbf{f}, \mathbf{e})$
- we'd like only to discriminate bad translations from the good ones
- assuming given $\mathbf{f}$ (conditional)

- $\phi(\mathbf{f}, \mathbf{e})$ features

- $\phi(\mathbf{f}, \mathbf{e})$ features
- $\Delta(\mathbf{e}, \mathbf{rf})$ quality measure of $\mathbf{e}$

- $\phi(\mathbf{f}, \mathbf{e})$ features
- $\Delta(\mathbf{e}, \mathbf{rf})$ quality measure of $\mathbf{e}$
- $D = \{\mathbf{e}_i, \mathbf{rf}_i\}$ tuning corpus

- $\phi(\mathbf{f}, \mathbf{e})$ features
- $\Delta(\mathbf{e}, \mathbf{rf})$ quality measure of $\mathbf{e}$
- $D = \{\mathbf{e}_i, \mathbf{rf}_i\}$ tuning corpus
- $\mathbf{e}^*_{\mathbf{f}} = \arg\max_{\mathbf{e}} score(\mathbf{f}, \mathbf{e})$ decision rule

- $\phi(\mathbf{f}, \mathbf{e})$ features
- $\Delta(\mathbf{e}, \mathbf{rf})$ quality measure of $\mathbf{e}$
- $D = \{\mathbf{e}_i, \mathbf{rf}_i\}$ tuning corpus
- $\mathbf{e}^*_{\mathbf{f}} = \arg\max_{\mathbf{e}} \lambda \cdot \phi(\mathbf{f}, \mathbf{e})$ decision rule

- $\phi(\mathbf{f}, \mathbf{e})$ — features
- $\Delta(\mathbf{e}, \mathbf{rf})$ — quality measure of $\mathbf{e}$
- $D = \{\mathbf{e}_i, \mathbf{rf}_i\}$ — tuning corpus
- $\mathbf{e}^*_\mathbf{f} = \arg\max_\mathbf{e} \lambda \cdot \phi(\mathbf{f}, \mathbf{e})$ — decision rule
- $\mathcal{L}(D, \{\mathbf{e}^*\})$ — loss

- $\phi(\mathbf{f}, \mathbf{e})$ features
- $\Delta(\mathbf{e}, \mathbf{rf})$ quality measure of $\mathbf{e}$
- $D = \{\mathbf{e}_i, \mathbf{rf}_i\}$ tuning corpus
- $\mathbf{e}^*_{\mathbf{f}} = \arg\max_{\mathbf{e}} \lambda \cdot \phi(\mathbf{f}, \mathbf{e})$ decision rule
- $\mathcal{L}(D, \{\mathbf{e}^*\})$ loss
- **Task**: find λ s.t. loss is minimized

Ideally, we'd like to optimize the expected loss

$$\sum_{\mathbf{f}} p(\mathbf{f})\Delta(\mathbf{e}^*_{\mathbf{f}})$$

- $p(\mathbf{f})$ unknown

Ideally, we'd like to optimize the expected loss

$$\sum_{\mathbf{f}} p(\mathbf{f}) \Delta(\mathbf{e}^*_{\mathbf{f}})$$

- $p(\mathbf{f})$ unknown

Assuming $\{\mathbf{f}_i\}$ are faithfully sampled from $p(\mathbf{f})$, optimize instead the *empirical* loss

$$\sum_{i} \Delta(\mathbf{e}^*_{\mathbf{f}_i})$$

- $\Delta(\mathbf{e}^*_{\mathbf{f}_i}) = \Delta(\arg\max_{\mathbf{e}} \lambda \cdot \phi(\mathbf{f}, \mathbf{e}))$
- for the MT measures we know the loss is, at least, non-convex, non-smooth and non-continuous

But BLEU is also a corpus measure, not sentence-decomposable.

We are left with:

$$\Delta(\{\mathbf{f}_i\}, \{\mathbf{e}^*_{\mathbf{f}_i}\})$$

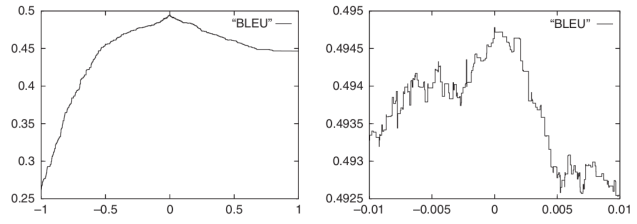
- non-convex, non-smooth and non-continuous

But BLEU is also a corpus measure, not sentence-decomposable.

We are left with:

$$\Delta(\{\mathbf{f}_i\}, \{\mathbf{e}^*_{\mathbf{f}_i}\})$$

- non-convex, non-smooth and non-continuous



But BLEU is also a corpus measure, not sentence-decomposable.

We are left with:

$$\Delta(\{\mathbf{f}_i\}, \{\mathbf{e}^*_{\mathbf{f}_i}\})$$

- non-convex, non-smooth and non-continuous
- does not split into sentences
- $\mathbf{e}^*$ is an implicit function of the search space, which is a function of w
  - ➡ beam search
  - ➡ pruning
- $\Rightarrow$ iterative training
  - ➡ updates of λ during training make the search space no longer correspond to the new weights
  - ➡ need to keep the search space up to date with periodic re-decode passes

But BLEU is also a corpus measure, not sentence-decomposable.

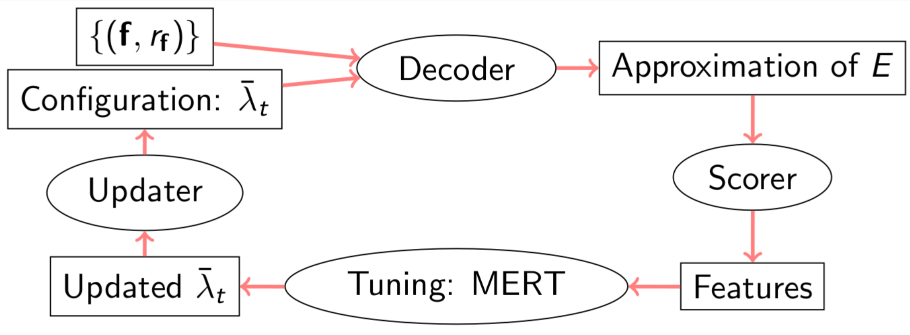
We are left with:

$$\Delta(\{\mathbf{f}_i\}, \{\mathbf{e}^*_{\mathbf{f}_i}\})$$

- non-convex, non-smooth and non-continuous
- does not split into sentences
- $\mathbf{e}^*$ is an implicit function of the search space, which is a function of w
  - ➡ beam search
  - ➡ pruning
- $\Rightarrow$ iterative training
  - ➡ updates of λ during training make the search space no longer correspond to the new weights
  - ➡ need to keep the search space up to date with periodic re-decode passes

approximations are inevitable

- **n-best lists**
  top-n highest scoring hypotheses according to the model score $w \cdot \phi$
- **lattices or hypergraphs**
  the underlying representations decoding runs on
  - ➡ these track all expanded hypotheses created while decoding
  - ➡ usually only the top-1 is used
  - ➡ can be dumped to extract other hypotheses (as model makes errors top-1 is not necessarily the best)


$\{(\mathbf{f}, r_{\mathbf{f}})\}$
Configuration: $\bar{\lambda}_t$
Decoder
Approximation of E
Scorer
Features
Tuning: MERT
Updated $\bar{\lambda}_t$
Updater

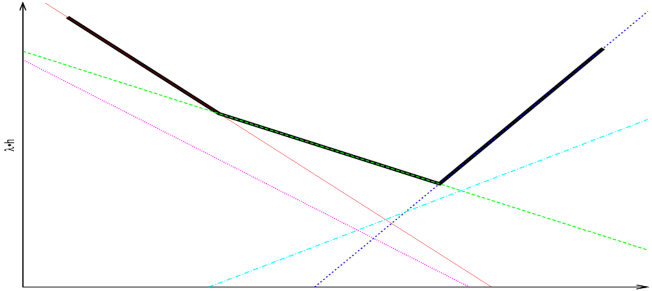
MERT proceeds in series of optimizations along directions $\bar{r}$:

$$\bar{\lambda} = \bar{\lambda}_0 + \gamma \bar{r}$$

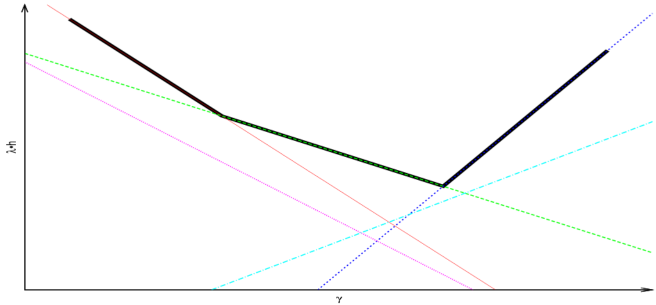
Optimal translation:

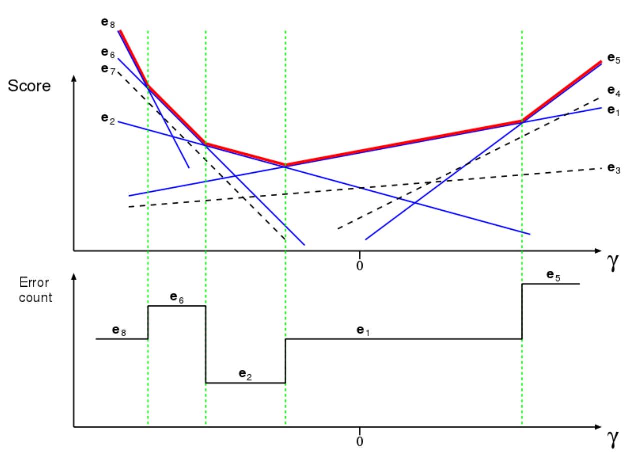
$$\tilde{\mathbf{e}}_{\mathbf{f}}(\gamma) = \arg\max_{\mathbf{e} \in E} \bar{\lambda} \cdot \bar{h}(\mathbf{e}, \mathbf{f}) = \arg\max_{\mathbf{e} \in E} \underbrace{\bar{\lambda}_0 \cdot \bar{h}(\mathbf{e}, \mathbf{f})}_{\text{intercept}} + \gamma \underbrace{\bar{r} \cdot \bar{h}(\mathbf{e}, \mathbf{f})}_{\textit{slope}}$$

- each translation hypothesis is associated with a line,
- **upper envelope**: dominating lines when $\bar{\lambda}$ is moved along $\bar{r}$

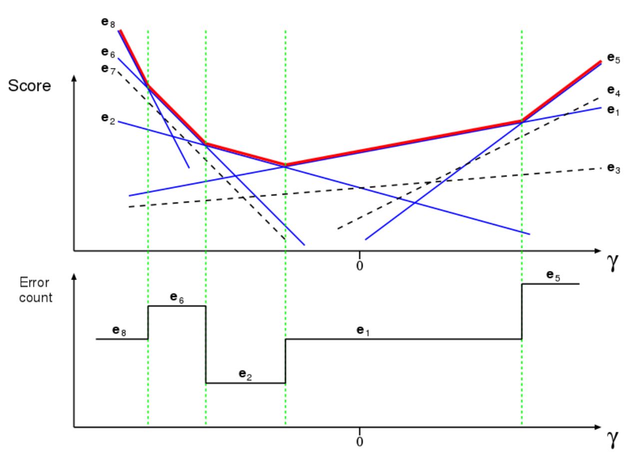


- γ-projections of intersections give intervals of constant optimal hypothesis
- optimal γ^* found by merging intervals for $\mathbf{f} \in F$ and scoring each
- update $\bar{\lambda} = \lambda_0 + \gamma^*_{i^*} \bar{r}_{i^*}$,
  where i^* is the index of the direction yielding the highest BLEU





[Macherey et al, 08]



[Macherey et al, 08]

- true irrespective of the loss function!
- finite number of values γ where the winning hypotheses changes
- $\Rightarrow$ much smaller than all γ!

- slow
- bad convergence of the cycle
  (typicaly $O(m)$ cycles for m-dimensional features)
- highly variable results, very sensible to initial conditions
- weights are difficult to trust
- optimisation landscape is very bumpy
- optimum not guaranteed
- good generalization performance not guaranteed

Tricks to improve optimization

- larger n-best lists (do not always help)
- restarts from different random λ_0
- restarts from promising points
- random directions (additionaly to axes)
- merges of n-best lists between iterations
- regularization/smoothing: average loss over neighbouring intervals $\Rightarrow$ more stable
- do it on lattices
- radical: change loss and/or optimization strategies
- many other tricks

Semiring $\mathbb{K} = \langle K, \oplus, \otimes, \bar{0}, \bar{1} \rangle$:

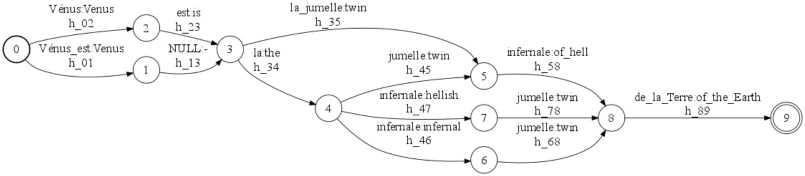
- $\langle K, \oplus, \bar{0} \rangle$ is a commutative monoid with identity element $\bar{0}$:
  - $a \oplus (b \oplus c) = (a \oplus b) \oplus c$
  - $a \oplus b = b \oplus a$
  - $a \oplus \bar{0} = \bar{0} \oplus a = a$
- $\langle K, \otimes, \bar{1} \rangle$ is a monoid with identity element $\bar{1}$
- $\otimes$ distributes over $\oplus$
  - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
  - $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$
- element $\bar{0}$ annihilates K
  - $a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$.

Examples

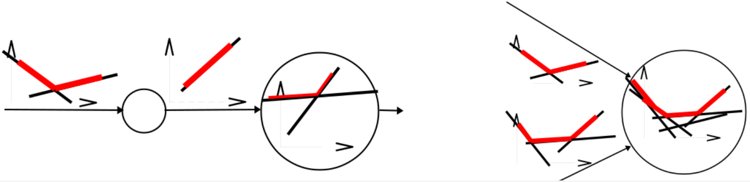
- $\langle \mathbb{R}, +, \times, 0, 1 \rangle$ – real semiring
- $\langle S, \Delta, \cap, \emptyset, \cup_i S_i \rangle$ – semiring of sets

source **fr**: Vénus est la jumelle infernale de la Terre

target **en**: Venus is Earth's hellish twin



- Decomposability of $\bar{h}(\mathbf{e}, \mathbf{f})$ into a sum of *local* features $h\_01, h\_02 \ldots$
- Envelopes are distributed over nodes in the lattice



$$\mathbb{D} = \langle D, \oplus, \otimes, \bar{0}, \bar{1} \rangle$$

Host set:

- a line: $d_y + d_s \cdot x$ (hypothesis)
- set of lines d_i: $d = \{d_{i,y} + d_{i,s} \cdot x\}$ (set of hypotheses)
- set of sets d^k of lines: $D = \left\{\{d^k_{i,y} + d^k_{i,s} \cdot x\}\right\}$

Operations $\oplus$ and $\otimes$:

- for $d^1, d^2 \in D$
- $d^1 \oplus d^2 = \text{env}(d^1 \cup d^2)$
- $d^1 \otimes d^2 = \text{env}(\{(d^1_{i.y} + d^2_{j.y}) + (d^1_{i.s} + d^2_{j.s}) \cdot x | \; \forall d^1_i \in d^1, d^2_j \in d^2\})$

Unities:

- $\bar{0} = \emptyset$
- $\bar{1} = \{0 + 0 \cdot x\}$

Each arc in the FST carries:

- target word a
- vector $\bar{h}(a, \mathbf{f})$ of local features associated with a
- singleton set containing line d with
  - slope $d_s = (\bar{r} \cdot \bar{h}(a, \mathbf{f}))$
  - y-intercept $d_y = (\bar{\lambda}_0 \cdot \bar{h}(a, \mathbf{f}))$

Weight of a candidate translation path $\mathbf{e} = e_1 \ldots e_\ell$:

$$w(\mathbf{e}) = \bigotimes_{i=1}^{\ell} w(e_i) = \{\bar{\lambda}_0 \cdot \sum_{i=1}^{\ell} \bar{h}(e_i, \mathbf{f}) + (\bar{r} \cdot \sum_{i=1}^{\ell} \bar{h}(e_i, \mathbf{f})) \cdot x\}$$

Upper envelope of all the lines (hypotheses):

$$\mathrm{env}(\bigcup_{\mathbf{e}} w(\mathbf{e})) = \bigoplus_{\mathbf{e}} w(\mathbf{e}) = \bigoplus_{\mathbf{e}} \bigotimes_{i=1}^{\ell(\mathbf{e})} w(e_i).$$

Generic shortest distance algorithms over acyclic graphs calculate this.

- pluggable into FST toolkits
- can be generalized to hypergraphs (SCFG)