Human Language Technology: Applications to Information Access

Lesson 7c: Tuning phrase-based statistical MT system with MERT

November 17, 2016

EPFL Doctoral Course EE-724
Andrei Popescu-Belis
Idiap Research Institute, Martigny

Reminder

- Principle of statistical MT (using Bayes' theorem)
  - learn English **language model**: $P(e)$
  - learn (reverse) **translation model**: $P(f|e)$
  - **decode** source sentence: find most likely e given f

$$\text{argmax}_e\ P(e|f) = \text{argmax}_e\ (P(f|e)\ P(e))$$

- Decode f = find e which maximizes the product of 3 terms
  - probabilities of inverse phrase translations $P_{\text{tm}}(\underline{f}_i|\underline{e}_i)$
  - reordering model for each phrase, e.g. $d(\text{START}(\underline{f}_i) - \text{END}(\underline{f}_{i-1}) - 1)$
  - language model for each word $P_{\text{lm}}\ (e_k|e_1,\ldots, e_{k\text{-}1})$

Log-linear models

- The three terms can be weighted
  - no longer a Bayesian model, but empirically more efficient
- Decoding find the sentence that maximizes

$$\prod_{i=1..M} (P_{\mathrm{tm}}(\underline{f}_i | \underline{e}_i)^{\lambda_{\mathrm{tm}}} \cdot d(\mathrm{START}(\underline{f}_i) - \mathrm{END}(\underline{f}_{i-1}) - 1)^{\lambda_{\mathrm{rm}}}) \cdot \prod_{k=1..|\mathrm{e}|} P_{\mathrm{lm}} (e_k | e_1, \ldots, e_{k-1})^{\lambda_{\mathrm{lm}}}$$

which can be expressed as: $P_\Lambda(e|f) = \exp\left(\sum \lambda_i h_i(e)\right)$ with $h(..) = \log P(..)$

and is thus equivalent to maximizing the sum without the 'exp'

- More terms can be added, e.g. word count penalty (the 4th weight in default `moses.ini`), but also reverse translation probabilities, lexical translation probabilities, or other dense/sparse features
  - How do we choose the optimal weights λ_i ?

Definition of tuning

- *Training* = learn translation & language models,
  on large parallel & monolingual corpora
- *Decoding* = find sentence maximizing scoring function

- ***Tuning*** = optimize the weights of the scoring function
  - on a small held-out set (hopefully similar to test data)
    - NB: tuning on the training set leads to overfitting
  - for a given error metric = distance to reference translation
  - i.o.w. tune the weights so that the translations of the tuning set get closer to the reference translations
    - dramatically improves MT scores on unseen data

Formal view of tuning

- By definition, the best parameter set is:

$$\boldsymbol{\Lambda}_{\mathrm{opt}} = \mathrm{argmax}_{\boldsymbol{\Lambda}} \left(\sum_{i=1..S} P_{\boldsymbol{\Lambda}}(e_i | f_i)\right)$$

  - where $\boldsymbol{\Lambda} = (\Lambda_1, \ldots, \Lambda_M)$ is the set of parameters and there are S sentences in the tuning set

- If we have a reference translation for each sentence, we can replace $P_{\boldsymbol{\Lambda}}(e_i | f_i)$ with the error, i.e. distance to the reference, and minimize:

$$\boldsymbol{\Lambda}_{\mathrm{opt}} = \mathrm{argmin}_{\boldsymbol{\Lambda}} \left(\sum_{i=1..S} Error(r_i, \hat{e}_{\boldsymbol{\Lambda}, i}\right)$$

  - where $\hat{e}_{\boldsymbol{\Lambda}, i}$ is the best translation hypothesis from the list generated by beam search for sentence f_i with reference r_i

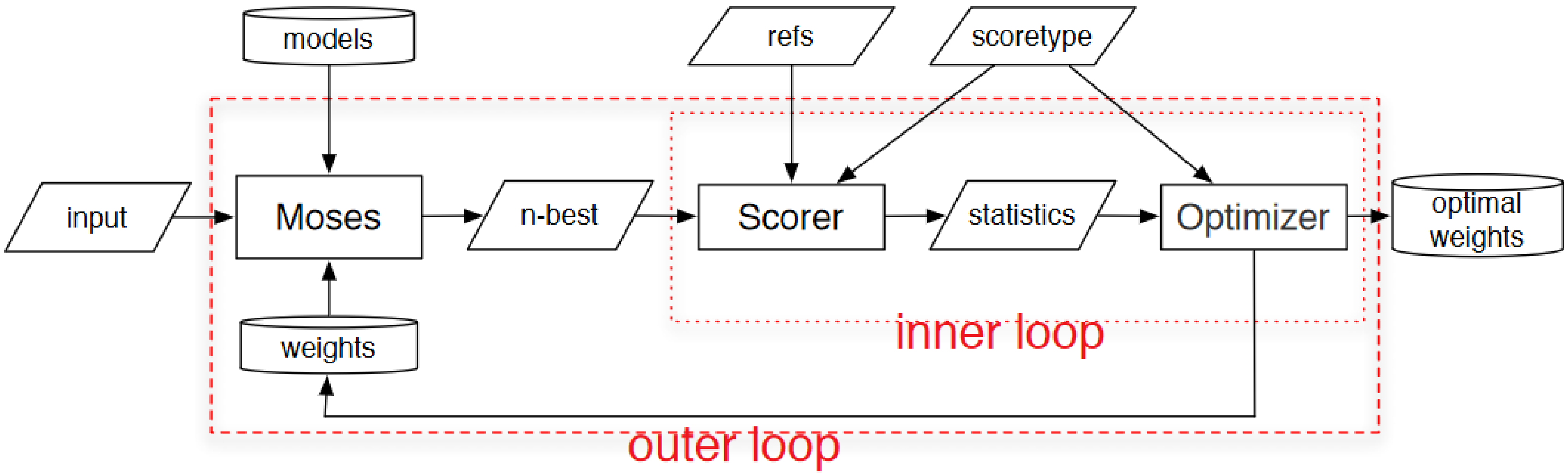
MERT: minimum error rate tuning

- Finding the best parameters $\mathbf{\Lambda}_{opt}$:
  - grid-based line optimization
    - optimize a Λ_k while keeping the others constant, then another one, etc.
  - large space, search is costly for fine-grained grids

- MERT optimization (Och, 2003)
  - take advantage of the fact the translation hypotheses can be enumerated, so varying Λ_k leads to a limited number of values
  - it is possible to calculate efficiently which of the parameters Λ_k would lead to the largest decrease of the total error when optimized
  - then pick this one, optimize it, and iterate
  - NB. MERT is a batch method: all data used for each iteration

Use of MERT in Moses

(from “Improved Minimum Error Rate Training in Moses” by Bertoldi N., Haddow B. and Fouet J.-B., 2009)



- Outer loop: translate (with new weights), then proceed to re-optimize using n-best lists
- Inner loop: score n-best lists, optimize one or more weights, then re-translate

Beyond MERT: a host of tuning methods
(reviewed by Neubig & Watanabe 2016)

- Evaluation measures to compare candidates vs. references
  – BLEU and variants; sentence-level vs. set-level
- Loss functions to optimize (on translation candidates vs. reference)
  – error of 1-best, softmax, risk, margin, ranking, min. squared error
- Optimization algorithm to use
  – MERT, gradient based methods, margin-based, linear regression, MIRA
- Nature/number of translation candidates to consider
  – k-best or lattice or forest, or output of forced decoding
- Several methods are implemented in Moses: MERT is still very popular

  ➔ find a small but representative tuning set, run `mert-moses.pl`
  (notice how the weights of the parameters in `moses.ini` have changed)

References

- Recent overview of tuning approaches
  - "Optimization for Statistical Machine Translation: A Survey", by Graham Neubig and Taro Watanabe, *Computational Linguistics*, 2016.
- MERT
  - "Minimum error rate training in statistical machine translation", by Franz Josef Och, *Proceedings of ACL,* 2003.
- MIRA: Margin Infused Relaxed Algorithm
  - originally a multiclass classification method (Crammer & Singer, JMLR 2003), adapted to MT
  - "Online Large Margin Training for Statistical Machine Translation", by Watanabe T. et al., *Proceedings of EMNLP-CoNLL,* 2007.
  - "Batch tuning strategies for statistical machine translation", by Colin Cherry and George Foster, *Proceedings of NAACL,* 2012.
- PRO: Pairwise rank optimization
  - "Tuning as ranking", by Mark Hopkins and Jonathan May, *Proceedings of EMNLP*, 2011.
- Moses implements several tuning methods, see manual
  - http://www.statmt.org/moses/?n=Moses.Baseline see "Tuning"
  - http://www.statmt.org/moses/?n=FactoredTraining.Tuning