Computing Lattice BLEU Oracle Scores for Machine Translation

Artem Sokolov & Guillaume Wisniewski & François Yvon
{firstname.lastname}@limsi.fr
LIMSI, Orsay, France

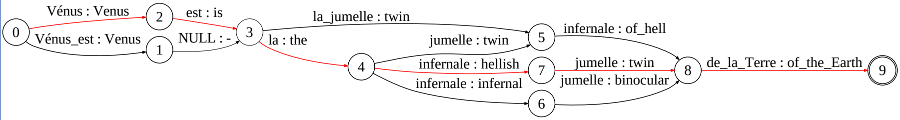
Usual translating of sentences f:

- best translation: $\mathbf{e}^* = \arg\max_{\mathbf{e} \in E(\mathbf{f})} score(\mathbf{e}|\mathbf{f})$
- e.g.: $score(\mathbf{e}|\mathbf{f}) = \lambda \cdot \bar{h}(\mathbf{e}, \mathbf{f})$ $\bar{h}(\mathbf{e}, \mathbf{f})$ – features, $\bar{\lambda}$ – params
- $E(\mathbf{f})$ – **search space** ⬅ interested in this

Example

- source: `Vénus est la jumelle infernale de la Terre`
- search space options:
  - ➡ list of n-best hypotheses

    ```
    Venus - the twin of hell of the Earth
    Venus is the infernal binocular of the Earth
    Venus is the hellish twin of the Earth
    ```
  - ➡ word lattice



- in reality SMT systems perform poorly
  - ➥ knowing why is difficult because of system complexity
- useful to know the best possible hypotheses
  - ➥ can help failure analysis

Usefulness of (oracle) translation

- Finding bottlenecks
  - ➡ if oracle is good – why the system couldn't return it?
  - ➡ if oracle is bad – why doesn't E contain better ones?
- Learning
  - ➡ update towards the best hypothesis

In this work

- interested in finding best hypotheses contained **in lattices**
- two new methods, comparison with 2 existing methods

- by best hypothesis in E we mean highest BLEU translation

$$n\text{-}BLEU(\{\mathbf{e}_i\},\{\mathbf{r}_i\}) = BrevityPenalty \cdot \left(\prod_{m=1}^{n} p_m\right)^{1/n}$$

  - ➡ $\mathbf{e}_i$ – hypotheses, $\mathbf{r}_i$ – references, p_m – clipped precisions
- need some sentence-level approximation BLEU'

Exact oracle finding with sent-BLEU'

- in n-best list – straight-forward
- in word lattice – **NP-hard** even for 2-BLEU [Leusch et al., 2008]
  - ➥ so, approximations are needed ⬅ focus of this presentation

Oracle decoding on lattice L for reference $\mathbf{r_f}$

$$\boldsymbol{\pi}^*(\mathbf{f}) = \underset{\boldsymbol{\pi}\in\{\text{paths on } L\}}{\arg\max}\; BLEU'(\mathbf{e}_{\boldsymbol{\pi}}, \mathbf{r_f}).$$

Language Model Oracle [Li & Khudanpur, 2009] LM

- n-gram language model A_{LM} built on reference $\mathbf{r}$
- find most probable hypothesis

$$\pi^*_{LM}(\mathbf{f}) = \texttt{ShortestPath}(L \circ A_{LM}(\mathbf{r_f}))$$

Partial BLEU Oracle [Dreyer et al., 2007] PB/PBℓ

- for partial path π let its weight be:

$$-\log BLEU'(\pi) = -\frac{1}{4} \sum_{m=1..4} \log p_m$$

- hypothesis h_1 is prefered to h_2 if $\log BLEU(h_1) > \log BLEU(h_2)$
  - ➡ and same 3-word history or PB
  - ➡ same 3-word history and same length PBℓ
- find shortest path

$$\pi^*_{PB}(\mathbf{f}) = \texttt{ShortestPath}(L)$$

Problems with existing approaches

1. LM
   - ➡ distant relation to BLEU
   - ➡ low quality
2. PB
   - ➡ approximate search (recombination heuristics)
   - ➡ can be resource-greedy (keeps many hypotheses until the final state)
3. Both LM & PB ignore:
   - ➡ precision clipping
   - ➡ brevity penalty (except PBℓ)
   - ➡ corpus nature of BLEU

New oracles

- partially take clipping, brevity and corpus nature into account
- produce better hypotheses and often faster

Take-away message

- although *true* BLEU oracles are NP-hard...
- we can find *approximate* oracles:
  - very easy to calculate
  - very high BLEU

Linear approximation of corpus-BLEU

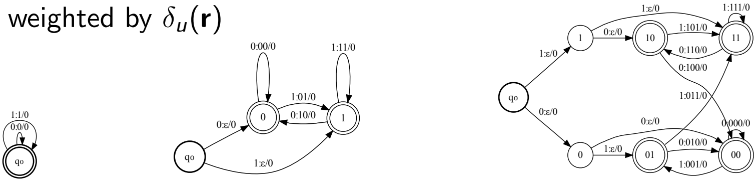
Linear approximation of BLEU [Tromble et al., 2008]

$$\text{lin}\,BLEU(\boldsymbol{\pi}) = \theta_0\,|\mathbf{e}_{\boldsymbol{\pi}}| + \sum_{n=1}^{4} \theta_n \sum_{u \in \Sigma^n} c_u(\mathbf{e}_{\boldsymbol{\pi}})\delta_u(\mathbf{r})$$

- $c_u(\mathbf{e})$ – number of occurences of u in $\mathbf{e}$,
- $\delta_u(\mathbf{r})$ – indicator of presence of u in $\mathbf{r}$.
- parameters: $\theta_0 = 1$ ($\sim$brevity), $\theta_n = -\frac{1}{4p \cdot r^{n-1}}$ (n-gram discounts)

'de Bruijn transducers' Δ_n

- each Δ_n contains all n-grams as output symbols
- arcs weighted by $\delta_u(\mathbf{r})$



Δ_n automata for alphabet $\{0, 1\}$ and $n = 1 \ldots 3$

Composition effect of Δs

- $L \circ \Delta_n$ produces n-gram lattice
- each n-gram arc weighted by θ_n, if the n-gram was in reference

Find LB oracle translation for 4-BLEU

- weight lattice L with θ_0
- weight Δ_i with θ_i
- find shortest path

$$\pi^*_{LB} = \mathtt{ShortestPath}(L \circ \Delta_1 \circ \Delta_2 \circ \Delta_3 \circ \Delta_4).$$

- so far all oracles:
  - ➡ transformed and/or reweighted lattice L
  - ➡ called `ShortestPath`
- difficult to account for clipping (it depends on the resulting path)

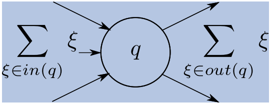
Integer Linear Programming Oracles

- how to take clipping into account?
- need to fine-tune path weights
- reformulate 1-BLEU shortest path problem as an ILP problem:

$$\arg\max_{\boldsymbol{\xi}} \sum_{i \in \{arcs\}} \xi_i \cdot (\Theta_1 \cdot \mathbb{1}_{i \in \mathbf{r}} - \Theta_2 \cdot \mathbb{1}_{i \notin \mathbf{r}})$$ maximize 1-BLEU

$$\sum_{\xi \in in(q_{fin})} \xi = 1, \quad \sum_{\xi \in out(q_{init})} \xi = 1$$ solution should be path

$$\sum_{\xi \in out(q)} \xi - \sum_{\xi \in in(q)} \xi = 0, \;\; q \in \{nodes\} \setminus \{q_{init}, q_{fin}\}$$



- arc indicator $\xi_i = \begin{cases} 1 & \text{if arc } i \text{ is active in solution,} \\ 0 & \text{otherwise,} \end{cases}$

New variable: Clipped word counter

$$\gamma_w = \min\{\sum_{\xi \in \{\text{edges with } w\}} \xi, \quad \# \text{ of } w \text{ in } \mathbf{r}\}$$

ILP problem with clipping

$$\arg\max_{\boldsymbol{\xi}, \gamma_w} \quad \Theta_1 \cdot \sum_{w \in \{words\}} \gamma_w - \Theta_2 \cdot \left(\sum_{i \in \{arcs\}} \xi_i - \sum_{w \in \{words\}} \gamma_w \right)$$

$$\gamma_w \geq 0, \gamma_w \leq c_w(\mathbf{r}), \gamma_w \leq \sum_{\xi \in \{\text{edges carring } w\}} \xi$$

$$\sum_{\xi \in in(q_{fin})} \xi = 1, \sum_{\xi \in out(q_{init})} \xi = 1$$

$$\sum_{\xi \in out(q)} \xi - \sum_{\xi \in in(q)} \xi = 0, \; q \in \{nodes\} \setminus \{q_{init}, q_{fin}\}$$

- many ILP solvers available
- we used Gurobi

- can be done without 3rd-party solvers
- Lagrangian relaxation to deal with clipping

Primal

$$\arg\min_{\boldsymbol{\xi} \in \{paths\}} - \sum_{i \in arcs} \xi_i \cdot \theta_i$$

$$\sum_{\xi \ \ with \ \ w} \xi \leq c_w(\mathbf{r}), w \in \mathbf{r}$$

Dual

$$\arg\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda})$$

$$\lambda_w \succeq 0, w \in \mathbf{r}$$

Where

$$\mathcal{L}(\boldsymbol{\lambda}) = \min_{\boldsymbol{\xi}} \left\{ - \sum_{i \in \{arcs\}} \xi_i \cdot \theta_i + \sum_{w \in \mathbf{r}} \lambda_w \cdot \left(\sum_{\xi \ \ with \ \ w} \xi - c_w(\mathbf{r}) \right) \right\}$$

$$\mathcal{L}(\boldsymbol{\lambda}) = \min_{\boldsymbol{\xi}} \left\{ -\sum_{i\in\{arcs\}} \xi_i\theta_i + \sum_{w\in\mathbf{r}} \lambda_w \left(\sum_{\xi \text{ with } w} \xi - c_w(\mathbf{r}) \right) \right\}$$

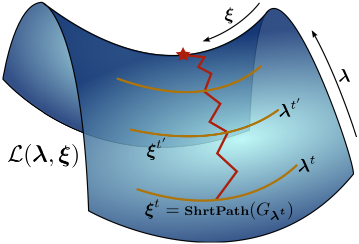
$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{\xi}} \sum_{i \in \{arcs\}} \xi_i \left(\lambda_{w(\xi_i)} - \theta_i \right)$$ ⬅ just find shortest path

$$\frac{\partial \mathcal{L}(\boldsymbol{\lambda})}{\partial \lambda_w} = \sum_{\boldsymbol{\xi}\in\Omega(w)\cap\boldsymbol{\xi}^*} \xi - c_w(\mathbf{r})$$ ⬅ update $\boldsymbol{\lambda}$ towards max $\mathcal{L}(\boldsymbol{\lambda})$

Incremental Constraint Enforcing

cf. [Rush et al., 2010]

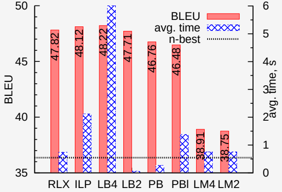
$\forall w, \lambda_w^{(0)} \leftarrow 0$
for $t = 1 \rightarrow T$ **do**
  $\boldsymbol{\xi}^{*(t)} = \arg\min_{\boldsymbol{\xi}} \sum_i \xi_i \cdot (\lambda_{w(\xi_i)} - \theta_i)$
  if *all clipping constraints are enforced*
  then optimal solution found
  else for $w \in \mathbf{r}$ **do**
    $n_w \leftarrow$ n. of occurrences of w in $\boldsymbol{\xi}^{*(t)}$
    $\lambda_w^{(t)} \leftarrow \lambda_w^{(t)} + \alpha^{(t)} \cdot (n_w - c_w(\mathbf{r}))$
    $\lambda_w^{(t)} \leftarrow \max(0, \lambda_w^{(t)})$



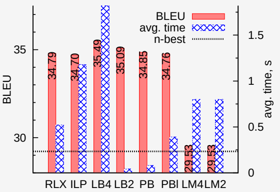
- this was for 1-BLEU
- for n-BLEU run on $L \circ \Delta_n$

Experiments: Performance for two decoders

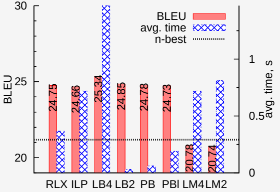
N-code



(a) fr→en (test: 27.88)

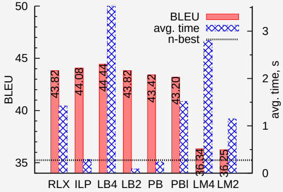


(b) de→en (test: 22.05)

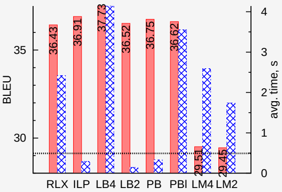


(c) en→de (test: 15.83)

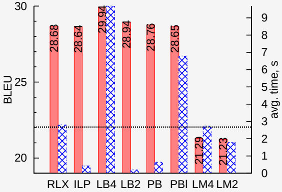
Moses



(d) fr→en (test: 27.68)



(e) de→en (test: 21.85)



(f) en→de (test: 15.89)

Results

- new oracle decoders find better hypotheses (clipping/brevity/corpus)
- ...and do it faster
- optimization for 2-BLEU is sufficient to obtain good 4-BLEU

Main Conclusion

➡ **lattices contain excellent translations**

- **+5-10** points to *n*-best list oracles
- **+20** points above results of state-of-the-art models

➡ **SMT has serious scoring problems**

- linear models not flexible enough?
- more features needed?
- training flawed?

Future Work

➡ non-linear score func.

➡ rich features

➡ discriminative training with lattice oracles

Thank you for your attention!

Questions?

Dreyer, M., Hall, K. B., & Khudanpur, S. P. (2007).
Comparing reordering constraints for SMT using efficient BLEU oracle computation.
In *Proc. of the Workshop on Syntax and Structure in Statistical Translation* (pp. 103–110). Morristown, NJ, USA.

Leusch, G., Matusov, E., & Ney, H. (2008).
Complexity of finding the BLEU-optimal hypothesis in a confusion network.
In *Proc. of the 2008 Conf. on EMNLP* (pp. 839–847). Honolulu, Hawaii.

Li, Z. & Khudanpur, S. (2009).
Efficient extraction of oracle-best translations from hypergraphs.
In *Proc. of Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the ACL, Companion Volume: Short Papers* (pp. 9–12). Morristown, NJ, USA.

Rush, A. M., Sontag, D., Collins, M., & Jaakkola, T. (2010).
On dual decomposition and linear programming relaxations for natural language processing.
In *Proc. of the 2010 Conf. on EMNLP* (pp. 1–11). Stroudsburg, PA, USA.

Tromble, R. W., Kumar, S., Och, F., & Macherey, W. (2008).
Lattice minimum bayes-risk decoding for statistical machine translation.
In *Proc. of the Conf. on EMNLP* (pp. 620–629). Stroudsburg, PA, USA.

Implementation

- OpenFST toolbox
  - ➡ works with **any** semiring (PB required semiring of sets)
  - ➡ proven and well optimized `ShortestPath` algorithms
- Gurobi ILP solver